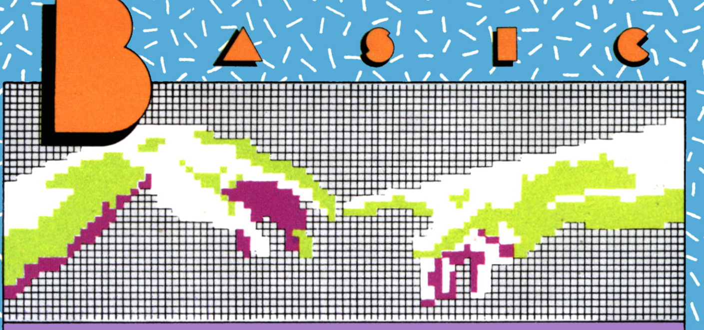


PROGRAMAS de APLICACIONES en BASIC

MATEMÁTICA • FÍSICA, QUÍMICA Y BIOLOGÍA • GEOGRAFÍA E HISTORIA
LENGUA E IDIOMAS • BASES DE DATOS

E. Lowy Frutos – A.E. Gallego Palomero
S. Mansilla Romo



El Corte Inglés

EQUIPO DE AUTORES

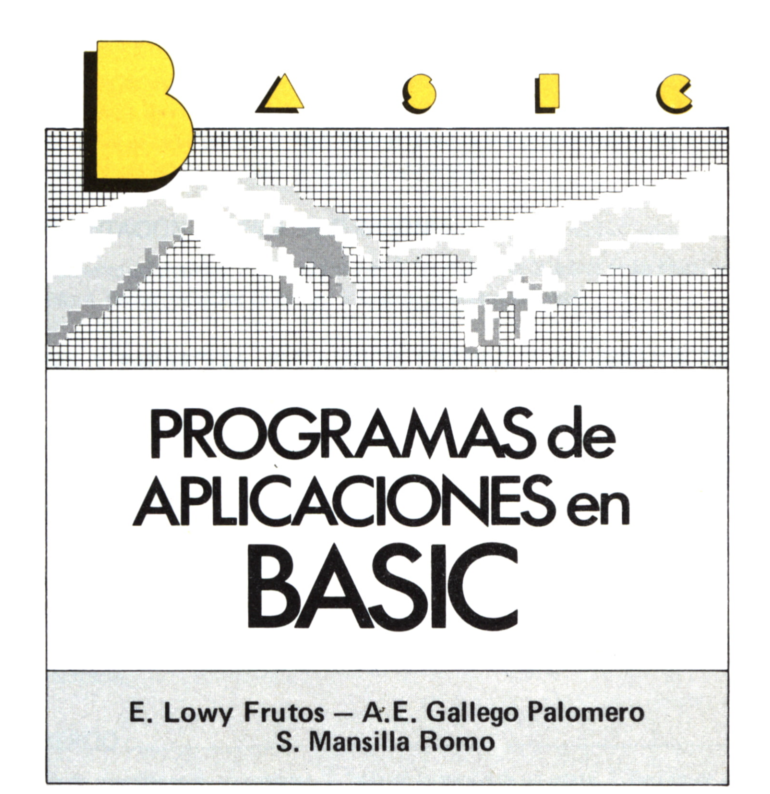
Ernesto Lowy Frutos
Agregado de Física y Química de I. B.

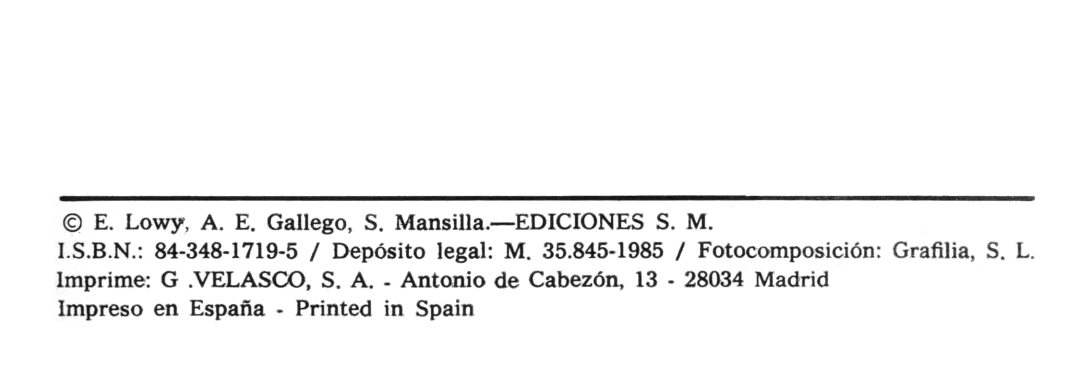
A. Enrique Gallego Palomero
Agregado de Matemáticas de I. B.

Serafín Mansilla Romo
Licenciado en Matemáticas

EQUIPO EDITORIAL

Coordinación: José Luis Robles Cid
Maqueta: José Ugarte
Dibujos: Modesto Arregui
Fotografías: J. M. Navia, Yolanda Álvarez, Javier Calbet
Portada: Alfonso Ruano y José Luis Cortés


BASIC
PROGRAMAS de APLICACIONES en BASIC
E. Lowy Frutos – A.E. Gallego Palomero
S. Mansilla Romo



I.S.B.N.: 84-348-1719-5 / Depósito legal: M. 35.845-1985 / Fotocomposición: Grafilia, S. L.
Imprime: G .VELASCO, S. A. - Antonio de Cabezón, 13 - 28034 Madrid
Impreso en España - Printed in Spain

Índice

PRESENTACIÓN

La resolución de problemas con el microordenador reporta varias ventajas, unas de carácter general o formativo y otras de carácter específico.

Entre las primeras cabe destacar la adquisición de un *hábito de rigor lógico* en el enfoque y planteamiento de cualquier problema relacionado con determinada área de conocimiento o con la vida real. Este hábito se afianza según se va utilizando el lenguaje de programación con el que nos comunicamos con el microordenador, en cuanto que dicho lenguaje lleva a una estructuración del pensamiento, ante la necesidad de descomponer el problema en una serie de pequeños pasos perfectamente relacionados.

Entre las ventajas de carácter específico hay que señalar el hecho de que la actividad intelectual que se desarrolla en la resolución de un problema concreto con el microordenador exige un discernimiento más profundo y riguroso de los conceptos vinculados al problema. En este sentido, la utilización del microordenador ofrece un *método de estudio* y *un estilo de analizar situaciones* que se caracterizan por ser más exhaustivos. Por ejemplo, la resolución de un problema de electricidad con el microordenador lleva a estudiar previamente esta parte de la física con más interés y rigor que si la resolución se hiciera por métodos tradicionales.

Con este libro, PROGRAMAS DE APLICACIONES EN BASIC, se pretende ofrecer una panorámica de las posibilidades que brinda el microordenador en el estudio de las distintas asignaturas y en la resolución de problemas.

El análisis de los programas de aplicaciones incluidos en el libro suscitarán en el lector el interés por elaborar otros programas para resolver situaciones que se le vayan presentando. Esto contribuirá a afianzar el rigor lógico y a obtener resultados prácticos en el estudio de las distintas áreas de conocimiento.

LOS AUTORES

En este libro se han utilizado las instrucciones de gráficos del estándar MSX.

1. Introducción

1. Objetivos de las aplicaciones

Con las distintas aplicaciones incluidas en este libro pretendemos ofrecer una muestra de la gran variedad de problemas que se pueden resolver con un microordenador.

Por otra parte, muchas aplicaciones, sobre todo las más sencillas, pueden servir de ejercicios de programación. En este caso, sugerimos comprobar los programas que se elaboren ejecutándolos en el microordenador, y a continuación compararlos con los que transcribimos en el libro.

2. Algunas pautas para resolver problemas con el microordenador

La solución de cualquier problema en informática constituye un proceso de investigación y, como tal, no se puede establecer un conjunto de reglas fijas que aplicándolas conduzcan a una solución correcta. Sin embargo, ciertas pautas o consejos pueden ser útiles en determinadas ocasiones.

Resumiremos a continuación algunas de estas ideas que facilitan el trabajo.

• Definición del problema

Se trata de simplificar y aislar el problema en que estamos interesados, eliminando toda la información no necesaria, para utilizar así el menor número posible de conceptos.

• Análisis del problema

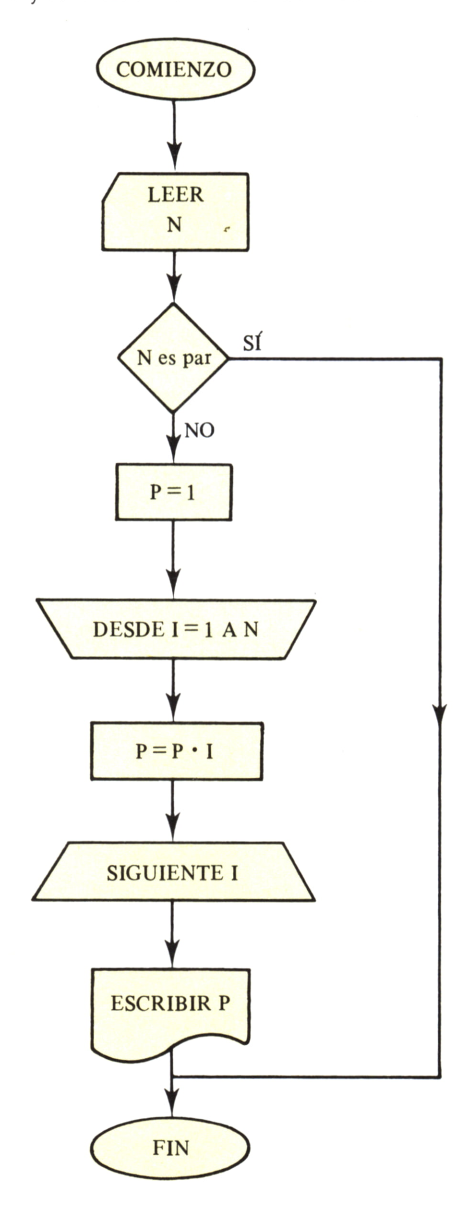
Una vez definido el problema debemos elegir una estrategia para llegar a una solución. Como se va a emplear un microordenador es necesario imponer una serie de condiciones, que vienen dadas por las instrucciones que éste es capaz de «entender» y realizar. Este proceso nos llevará así a un procedimiento, es decir, a un número finito de pasos (algoritmo), que permiten resolver el problema.

• Organigramas o diagramas de flujo

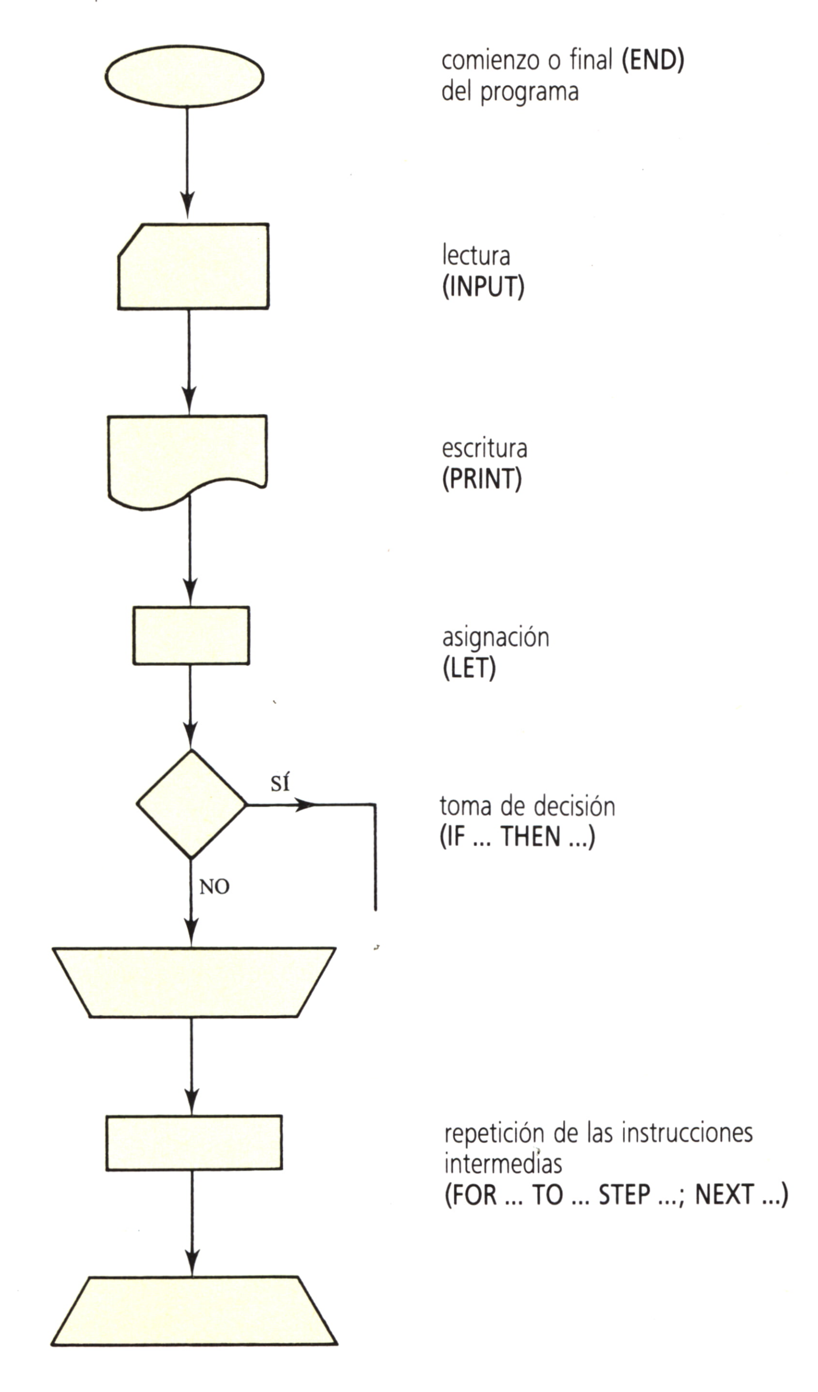
Los distintos pasos que supone el análisis de un problema se pueden indicar mediante esquemas gráficos. El conjunto de estos pequeños pasos vinculados entre sí y expresados gráficamente constituye lo que se llama *organigrama* o *diagrama de flujo.*

En un diagrama de flujo, las diferentes etapas a ejecutar se representan en recuadros en los que se incluyen las instrucciones específicas. El orden de ejecución de dichas etapas se indica mediante el sentido de las flechas que unen los recuadros.

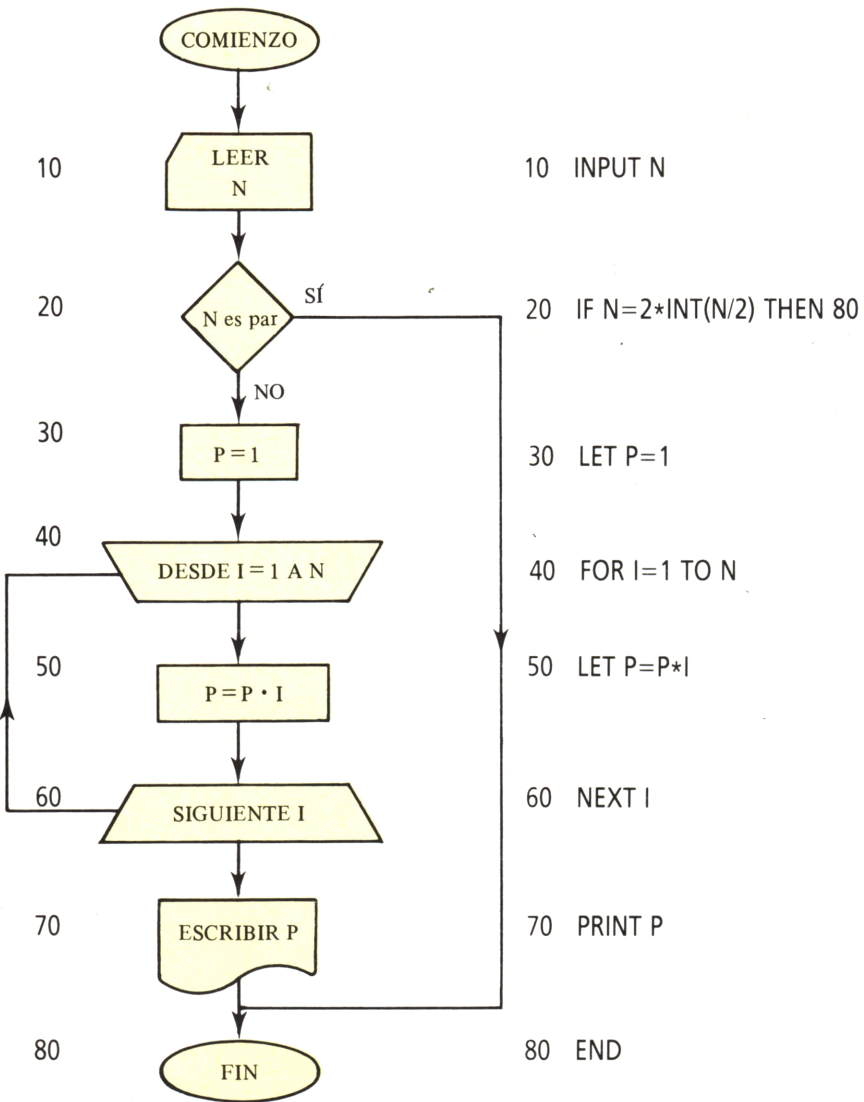
Como ejemplo, consideremos el diagrama de flujo correspondiente a un programa que lee un número N y calcula el producto de los N primeros números naturales si N no es par y en caso contrario no hace nada.



Los símbolos que se han utilizado son:



Como se pudo observar, a cada símbolo le corresponde una instrucción en BASIC (excepto a **COMIENZO**), por lo que es bastante fácil pasar del organigrama al correspondiente programa.



La instrucción **20** averigua si N es par o no. Para ello se utiliza la función **INT** (parte entera). Por ejemplo, si mediante **INPUT** se introduce N=25, se tendría:

```
IF N=2 * INT(25/2) THEN GOTO 80
```

El microordenador hace estos cálculos:

2 * INT(12.5)=2 * 12=24

y a continuación compara 25 con 24 (¿N=24?). Como no se cumple la igualdad (lo que supone que N no es par), ejecuta la instrucción siguiente a la **20.**

- **Elaboración de un programa**

Para concretar el procedimiento, pueden tenerse en cuenta los siguientes pasos:

a) Traducir todas las informaciones que manejamos a variables y conjuntos de variables.

b) Traducir todas las manipulaciones de datos a estructuras. Es decir, subdividir el programa en posibles bucles y subrutinas.
La utilización de subrutinas facilita una concepción y presentación estructurada de los programas. Cada programa puede considerarse, entonces, como una asociación de bloques de instrucciones, constituyendo uno de ellos el programa principal y los restantes, bloques anexos (las subrutinas o subprogramas).

c) Tratar de desarrollar el programa en un conjunto de bloques lo más completos posible en sí mismos. (En este sentido es útil usar **REM** para identificar las distintas partes.)

- **Corrección de errores en un programa**

Los programas construidos raramente funcionan la primera vez que se ejecutan.

El propio intérprete BASIC envía mensajes para señalar ciertos tipos de error (por ejemplo, de sintaxis), pero no detecta los errores lógicos cometidos.

Para solventarlos es necesario estudiar la ejecución del programa estableciendo etapas que se van analizando.

Esto puede conseguirse intercalando en el interior del programa instrucciones **STOP** (que detienen en un lugar determinado su ejecución), para posteriormente continuar hasta la siguiente etapa (siguiente **STOP**) utilizando la instrucción **CONT.**

En este proceso de comprobación podemos visualizar los valores que contienen las variables utilizando instrucciones **PRINT** en «modo directo». Así,

PRINT A(20) RETURN

permite conocer el contenido de la posición 20 de la variable A.

Debe tenerse en cuenta que **RUN** borra automáticamente el contenido previo de todas las variables en memoria cuando se inicia la ejecución del programa.

La instrucción **CLEAR** borra también el contenido de las variables, aunque no inicia la ejecución del programa.

Para evitar este efecto puede utilizarse la instrucción **GOTO** es «modo directo» que permite transferir la ejecución del programa a la línea que se desee.

• Verificación del programa

a) Para detectar errores en un programa, podemos ejecutarlo con datos para los cuales se conocen los resultados y compararlos a continuación con las soluciones obtenidas.

b) Si un programa contiene distintas bifurcaciones, debemos utilizar datos que permitan comprobar todas las opciones.

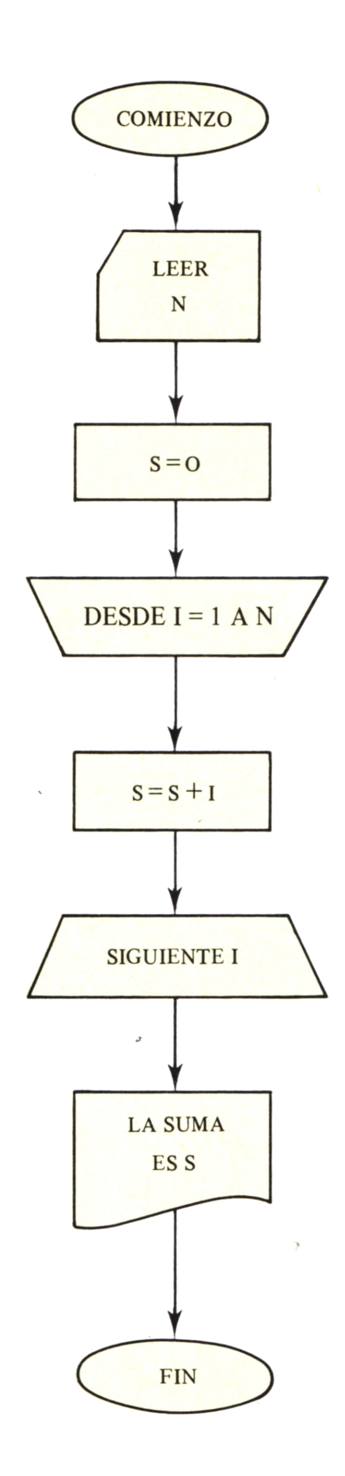
c) La técnica de construcción de un programa en diversos módulos facilita su verificación, ya que usando instrucciones de entrada y salida adicionales (que luego se eliminarán) permiten verificar los procesos intermedios.

d) Una técnica de depuración útil (sobre todo en programas cortos) es seguir los pasos del programa con lápiz y papel, simulando el trabajo del microordenador.

3. Ejercicios resueltos

1. Expresar en un organigrama el proceso que hay que seguir para sumar los N primeros números naturales.

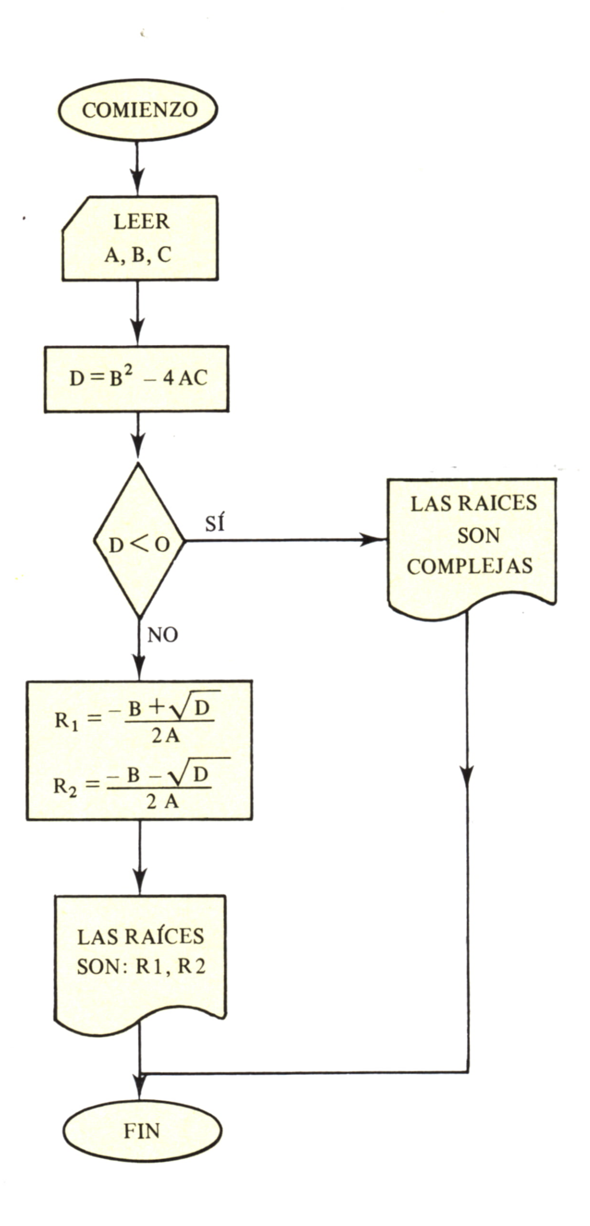
Solución



2. Obtener un organigrama en el que se indique el cálculo de las raíces reales de una ecuación de segundo grado.

Solución

Las letras A, B, C representan los coeficientes de la ecuación **$A\ x^2 + B\ x + C = 0$**

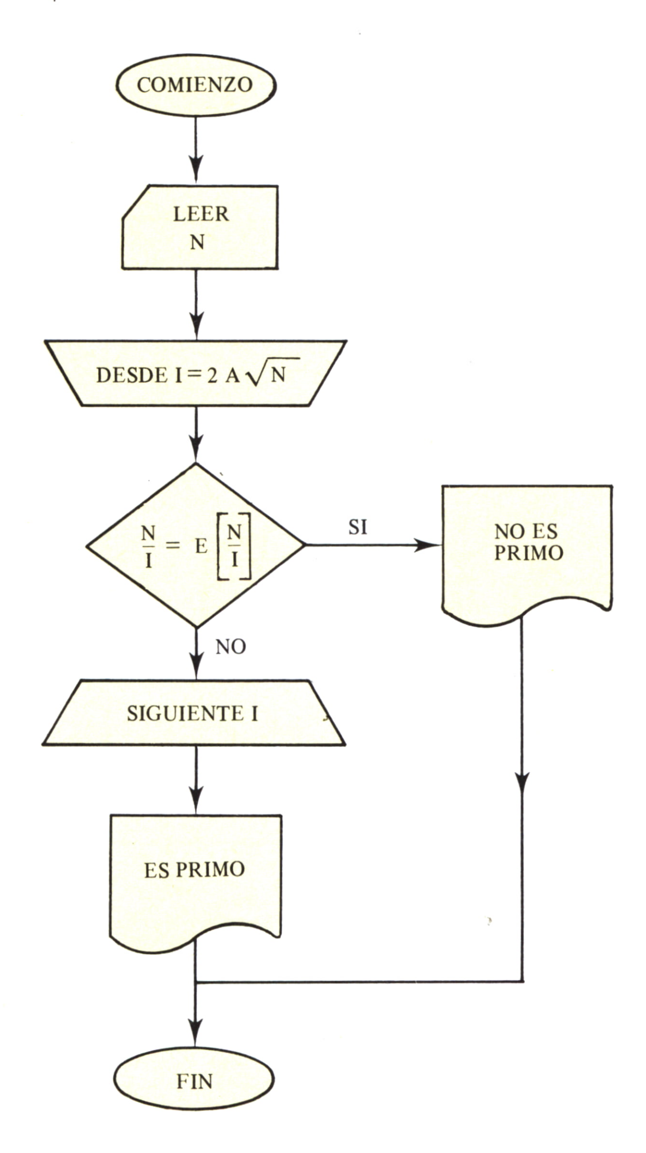


3. Expresar en un organigrama el proceso que hay que seguir para averiguar si un número es primo o no.

Solución

El símbolo E[x] representa la parte entera de X.

Para averiguar si N es primo tenemos que comprobar que no es divisible por los números que van de 2 a $\sqrt{N}$ (en realidad sólo por los que son primos):

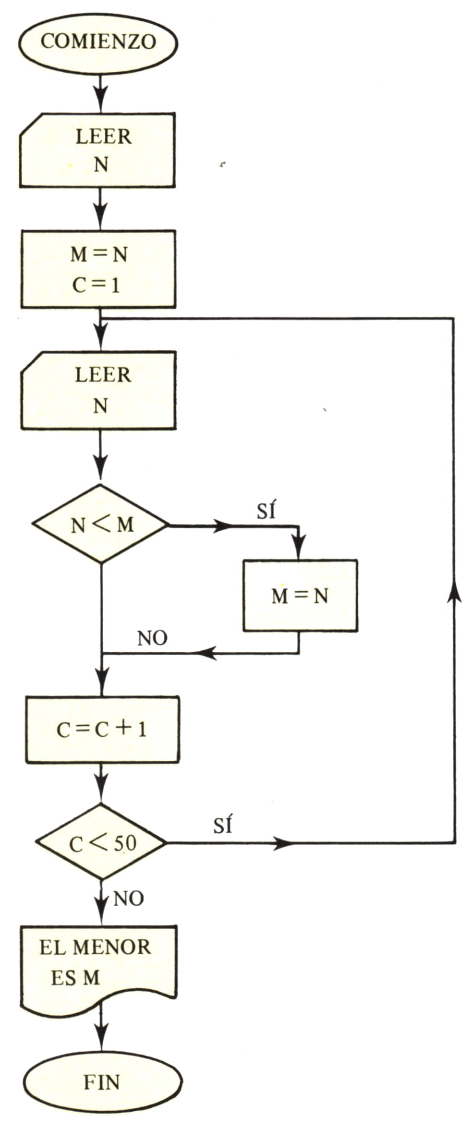


4. Realizar un organigrama para obtener el número menor de una lista de 50 números.

Solución

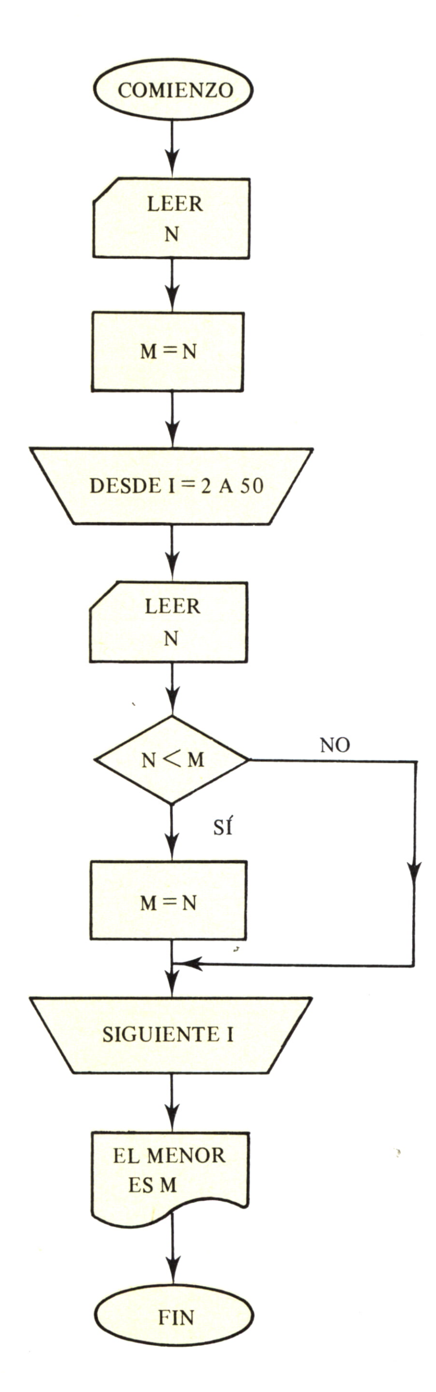
Las variables que se utilizan son:

— N (guarda el número leído)
— M (guarda el número menor hasta el momento)
— C (cuenta los números leídos)



Un organigrama sin contador:

Solución

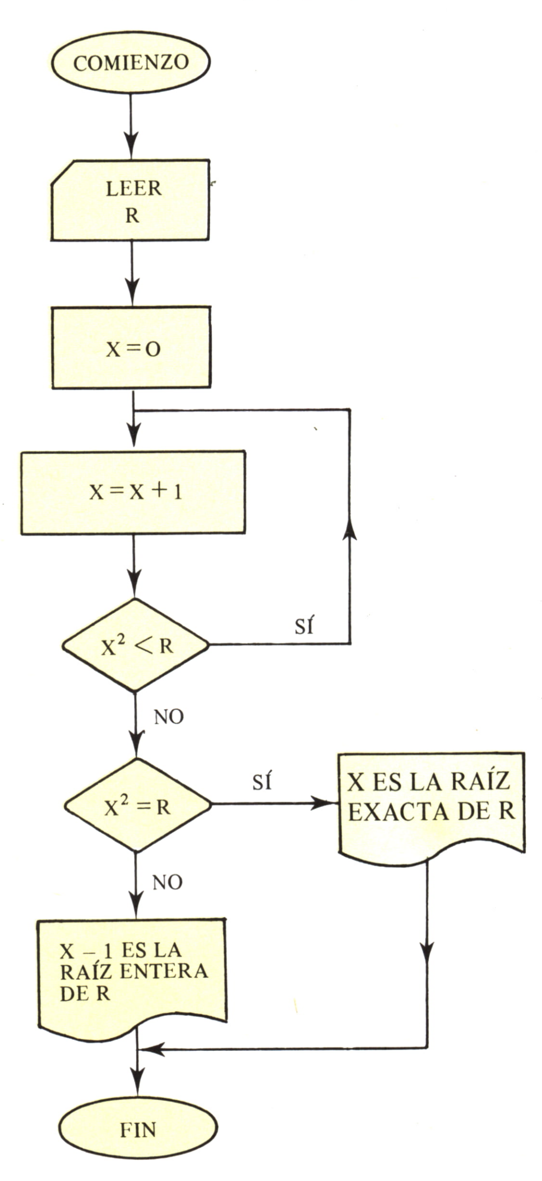


5. Encontrar un organigrama para calcular la raíz cuadrada entera de un número positivo.

Solución

La raíz cuadrada entera de un número R es un número x que elevado al cuadrado más el resto r es igual a R.

$$\sqrt{R} = x \Leftrightarrow x^2 + r = R$$



2. Dibujos a partir de la circunferencia

1. Resumen de las instrucciones específicas que se van a utilizar en este capítulo

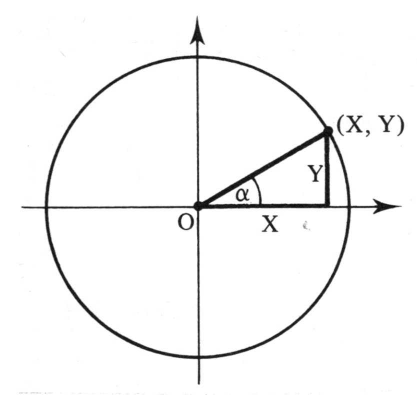
Las instrucciones específicas que se van a utilizar en este capítulo se resumen a continuación:

Instrucción	*Función*
SCREEN 2	Pasa a modo gráfico.
PSET (X, Y)	Imprime un punto en el pixel (X, Y).
LINE - STEP (X, Y)	Dibuja un segmento desde el último pixel impreso al pixel que se encuentra X posiciones a la derecha e Y posiciones hacia abajo.
CIRCLE (X, Y), R	Dibuja una circunferencia de centro el pixel (X, Y) y de radio R. (En los microordenadores IBM, MSX, etc., en lugar de circunferencia, dibuja una elipse.)
n GOTO n	Mantiene la imagen en la pantalla. (En los microordenadores MSX, presionar CTRL y STOP, para salir del bucle sin fin.)

2. Dibujo de una circunferencia «punto a punto»

Una circunferencia se puede dibujar directamente con la instrucción **CIRCLE.** Pero puede interesar dibujarla punto a punto, para lo cual es necesario determinar previamente las coordenadas de uno de los puntos de dicha circunferencia.

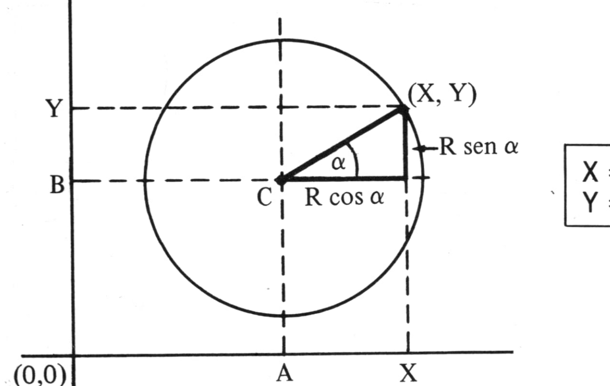
Suponiendo que el radio de la circunferencia es R y que el origen de coordenadas coincide con su centro, las coordenadas (X, Y) de un punto cualquiera de la misma, se obtienen partiendo de la definición de seno y coseno.



$$\text{sen } \alpha = \frac{Y}{R} \rightarrow Y = R \text{ sen } \alpha$$

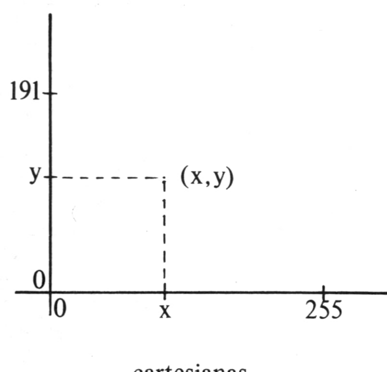
$$\cos \alpha = \frac{X}{R} \rightarrow X = R \cos \alpha$$

Si el centro C de la circunferencia se sitúa en el punto (A, B), las coordenadas (X, Y) se obtendrán sumando a R cos α el número A y a R sen α, el número B.

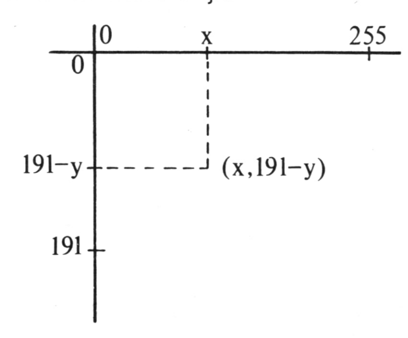


$$X = A + R \cos \alpha$$
$$Y = B + R \text{ sen } \alpha$$

Nota: En muchos microordenadores, como, por ejemplo, los MSX, las coordenadas cartesianas no coinciden con las coordenadas de la pantalla, pues los valores del eje vertical se ordenan de arriba abajo.



cartesianas



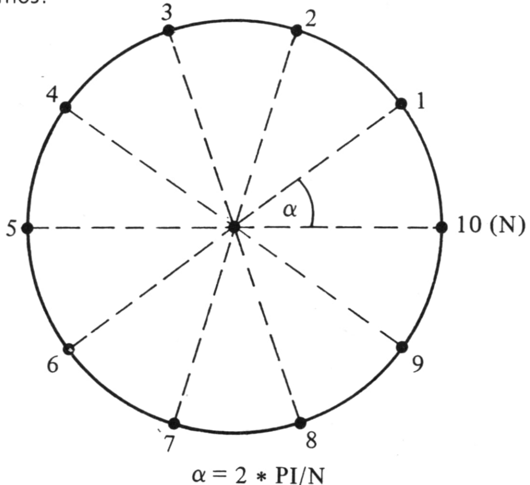
pantalla

La relación entre ambas coordenadas es la siguiente:

Al punto (X, Y) le corresponde el pixel (X, 191 – Y).

No obstante, al ser la circunferencia simétrica respecto del diámetro horizontal, no es necesario tener en cuenta esta relación.

Si decidimos dibujar la circunferencia con N puntos, tendremos que fijar previamente el valor del ángulo α necesario para pasar de un punto al siguiente. Este valor se calcula teniendo en cuenta que el ángulo completo correspondiente a toda la circunferencia es 360°, que equivale a 2 π radianes; luego si dibujamos N puntos, tendremos:



$\alpha = 2 * PI/N$

Es decir, el ángulo correspondiente al punto 1 valdrá 2 * PI/N * 1, el correspondiente al punto 2, 2 * PI/N * 2, ..., y el correspondiente al punto N, 2 * PI/N * N = = 2 * PI.

Luego, las coordenadas de los sucesivos puntos de la circunferencia se podrán determinar mediante las siguiente fórmulas:

X = A + R * COS (2 * PI/N * I)
Y = B + R * SIN (2 * PI/N * I)

siendo I = 1, 2, ..., N.

Para elaborar el programa que dibuje una circunferencia con una serie de puntos, hay que tener en cuenta las dos últimas fórmulas, fijando previamente los siguientes parámetros:

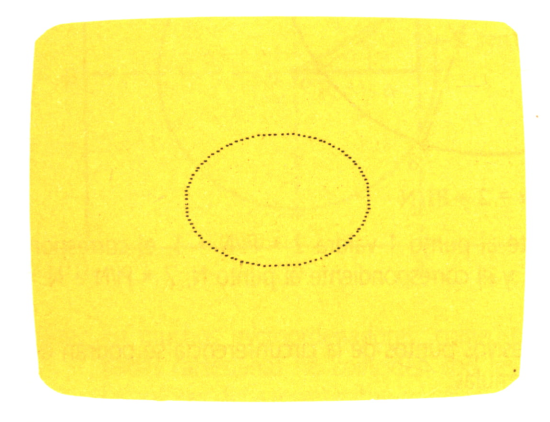
A, B: coordenadas del centro de la circunferencia
R: radio de la circunferencia
N: número de puntos

y tomar PI = 3.1416, con lo que 2 * PI = 6.2832.

Además, hay que definir la variable I, que tomará todos los valores enteros desde 1 a N, ambos inclusive. Así se obtiene el siguiente programa:

```
 10 INPUT "CENTRO"; A, B
 20 INPUT "RADIO"; R
 30 INPUT "NUMERO DE PUNTOS"; N
 40 SCREEN 2
 50 FOR I = 1 TO N
 60 X = A + R * COS (6.2832/N * I)
 70 Y = B + R * SIN (6.2832/N * I)
 80 PSET (X, Y)
 90 NEXT I
200 GOTO 200
```

Si lo ejecutamos para A = 127, B = 95, R = 50 y N = 100, obtenemos:



NOTA:
En esta fotografía y en las restantes de este capítulo no aparecen verdaderas circunferencias sino elipses. Ello es debido a que en los microordenadores MSX la separación de dos puntos consecutivos de la pantalla en posición horizontal es 1.4 veces mayor que la separación de los que se encuentran en posición vertical. Por esta misma razón, tampoco aparecen verdaderos polígonos regulares en las fotografías del apartado 4, *Polígonos regulares.*

(Ver *Dibujo de «verdaderas» circunferencias y de «verdaderos» arcos de circunferencia*, páginas 282 y 283 del libro **MSX. Programación, Gráficos, Colores y Música,** de los mismos autores).

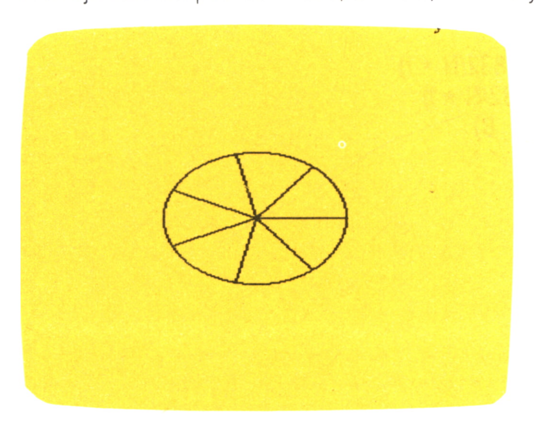
3. División de un círculo en N partes iguales

Para dividir una circunferencia en N partes iguales, la dibujaremos previamente con una instrucción **CIRCLE** y, a continuación, uniremos su centro con cada uno de los N puntos; así se obtendrán N sectores iguales.

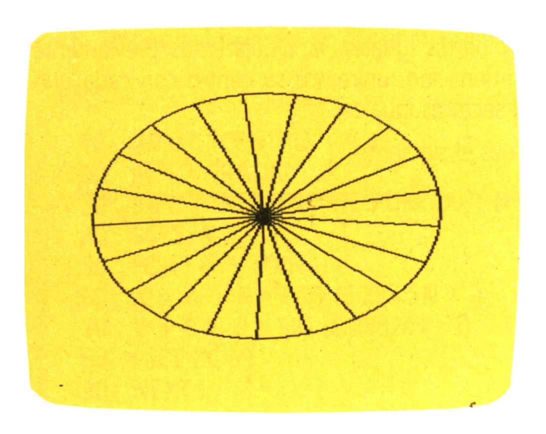
El programa que realiza esta división es el siguiente:

```
 10 INPUT "CENTRO"; A, B
 20 INPUT "RADIO"; R
 30 INPUT "PARTES"; N
 40 SCREEN 2
 50 CIRCLE (A, B), R
 60 FOR I = 1 TO N
 70 PSET (A, B)
 80 X = R * COS (6.2832/N * I)
 90 Y = R * SIN (6.2832/N * I)
100 LINE - STEP (X, Y)
110 NEXT I
200 GOTO 200
```

Si lo ejecutamos para A = 127, B = 95, R = 50 y N = 7, obtenemos:



Si lo ejecutamos para A = 127, B = 95, R = 95 y N = 21, tenemos:



4. Polígonos regulares

La división de la circunferencia en N partes iguales sugiere la obtención de polígonos regulares de N lados inscritos en la misma.

El siguiente programa dibuja polígonos regulares pero no la circunferencia en la que están inscritos. Esta circunferencia tiene su centro en el pixel (127, 95) y su radio es 80.

```
10  INPUT "NUMERO DE LADOS"; N
20  SCREEN 2
30  A = 127 + 80
40  B = 95
50  PSET (A, B)
60  FOR I = 1 TO N
70  X = 127 + 80 * COS (6.2832/N * I)
80  Y = 95 + 80 * SIN (6.2832/N * I)
90  LINE - STEP (X - A, Y - B)
100 A = X
110 B = Y
120 NEXT I
200 GOTO 200
```

Con la instrucción **10** se introduce el número de lados del polígono.

Las instrucciones **30, 40** y **50** imprimen un punto en el pixel (127 + 80, 95), donde comienza a dibujarse el polígono.

El bucle **60 – 120** dibuja los N lados.

Las instrucciones **70** y **80** calculan las coordenadas del siguiente punto de la circunferencia que hay que unir al A, B.

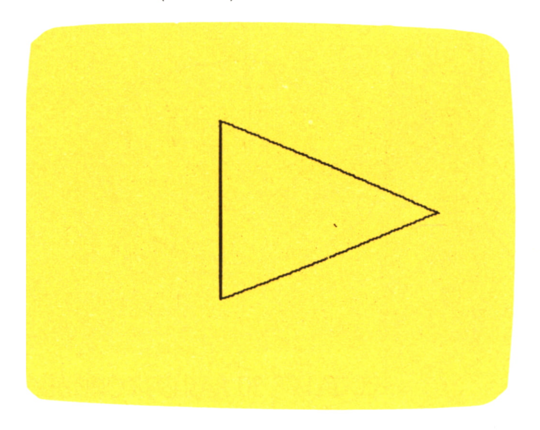
La instrucción **90 LINE - STEP (X – A, Y – B)** une los pixels (A, B) y (X, Y).

Las instrucciones **100** y **110** almacenan en A y B las coordenadas X, Y, calculadas por las instrucciones **70** y **80,** que serán el pixel de partida para dibujar el siguiente lado.

En resumen, (A, B) representan las coordenadas del origen de un lado y (X, Y), el punto final del lado.

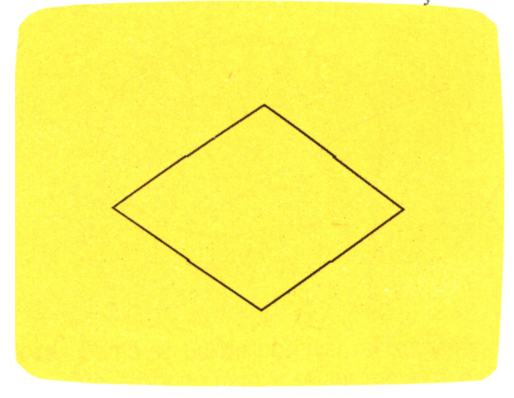
Si lo ejecutamos, tenemos

TRIÁNGULO (N = 3)

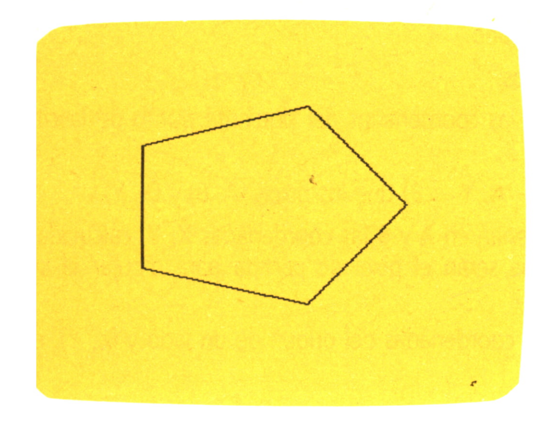




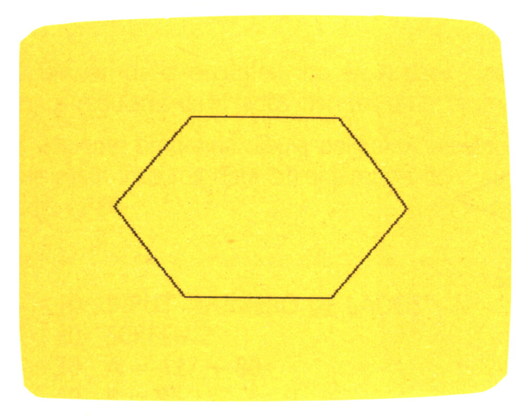
CUADRADO (N = 4)



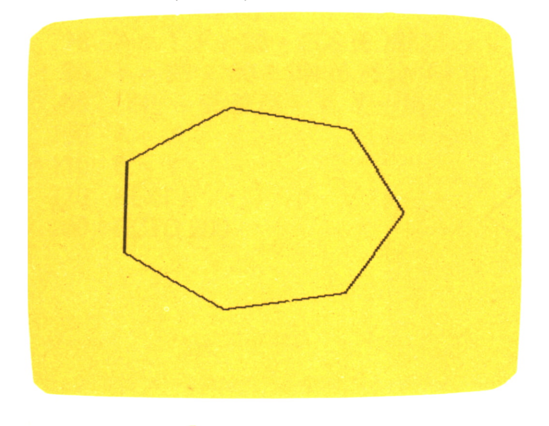
PENTÁGONO (N = 5)



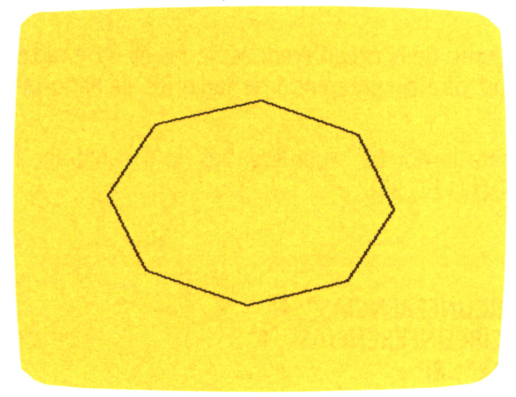
HEXÁGONO (N = 6)



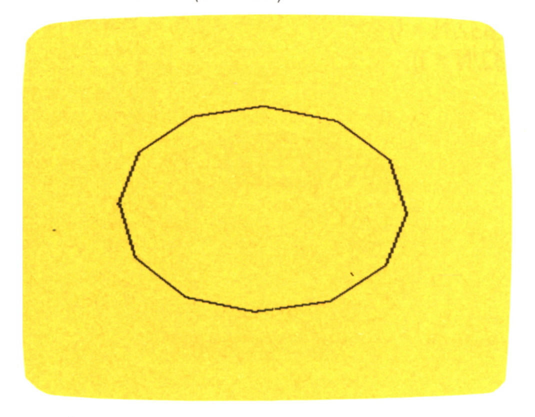
HEPTÁGONO (N = 7)



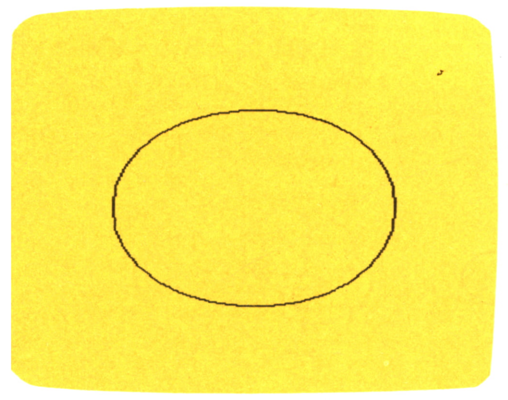
OCTÓGONO (N = 8)



DODECÁGONO (N = 12)



POLÍGONO REGULAR DE 30 LADOS



Nota: Como se puede observar, al aumentar el número de lados, el perímetro del polígono regular se aproxima cada vez más a la circunferencia.

5. Flor de circunferencias

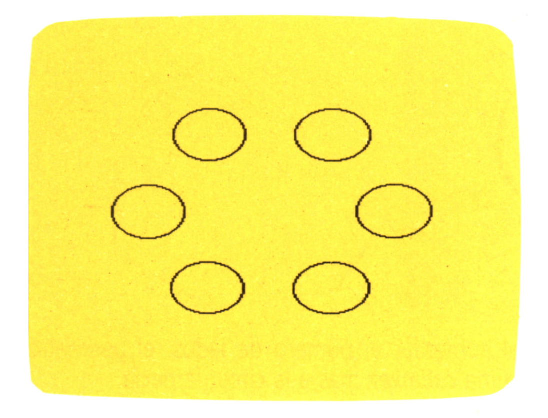
La flor de circunferencias es un conjunto de N circunferencias, todas ellas de radio R, cuyos centros determinan a su vez otra circunferencia de radio RP, llamado radio principal.

En el programa siguiente, que genera la flor de circunferencias, la circunferencia de radio RP tiene su centro en el pixel (127, 95).

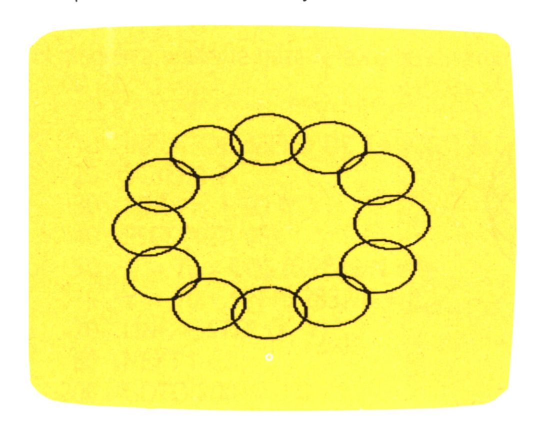
```
 10  INPUT "NUMERO DE CIRCUNFERENCIAS"; N
 20  INPUT "RADIO DE LAS CIRCUNFERENCIAS"; R
 30  INPUT "RADIO PRINCIPAL"; RP
 40  SCREEN 2
 50  FOR I = 1 TO N
 60  X = 127 + RP * COS (6.2832/N * I)
 70  Y = 95 + RP * SIN (6.2832/N * I)
 80  CIRCLE (X, Y), R
 90  NEXT I
200  GOTO 200
```

Cuando se ejecuta este programa, al introducir los datos hay que evitar que el dibujo se salga de la pantalla, para lo cual debe cumplirse R + RP ≤ 95.

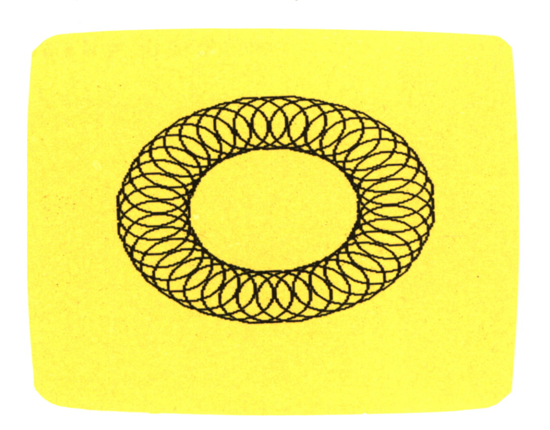
Flor para N = 6, R = 20 y RP = 67



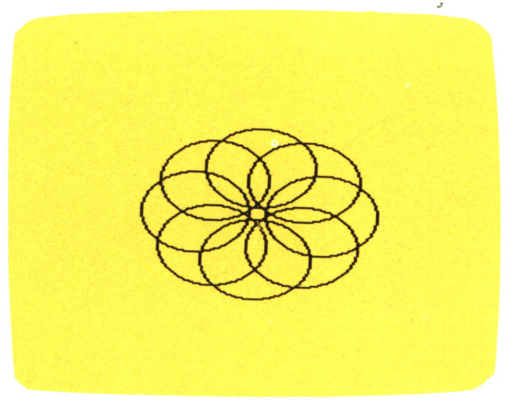
Flor para N = 12, R = 20 y RP = 67.



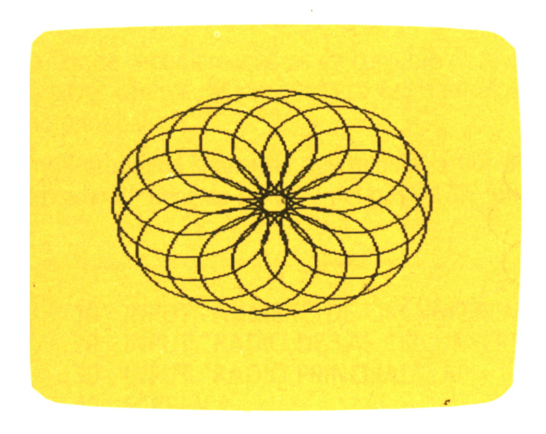
Flor para N = 40, R =20 y RP = 67.



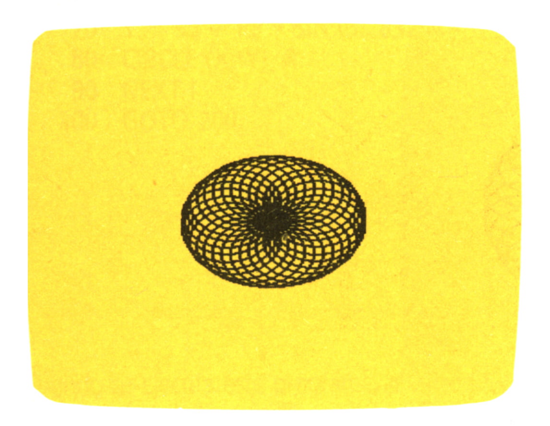
Flor para N = 8, R = 30 y RP = 35.



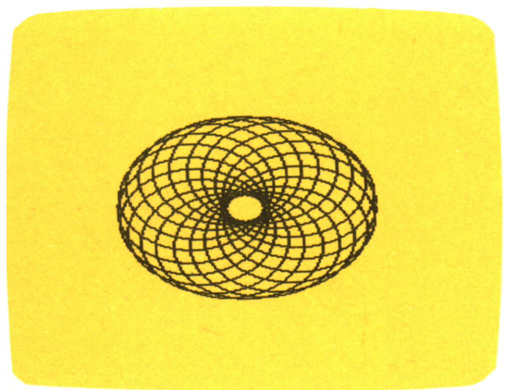
Flor para N = 16, R = 40 y RP = 47.



Flor para N = 30, R = 25 y RP = 25.



Flor para N = 20, R = 40 y RP = 30.

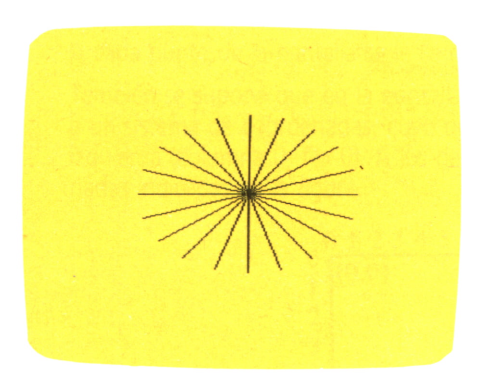


6. Rayos desde el centro de una circunferencia

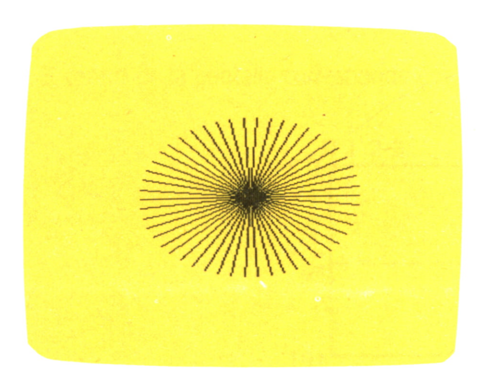
El programa siguiente traza N rayos de longitud 70, que parten del pixel (127, 95).

```
 10 INPUT "NUMERO DE RAYOS"; N
 20 SCREEN 2
 30 FOR I = 1 TO N
 40 PSET (127, 95)
 50 X = 70 * COS (6.2832/N * I)
 60 Y = 70 * SIN (6.2832/N * I)
 70 LINE - STEP (X, Y)
 80 NEXT I
200 GOTO 200
```

Para N = 20 se obtiene:



Para N = 50, tenemos:

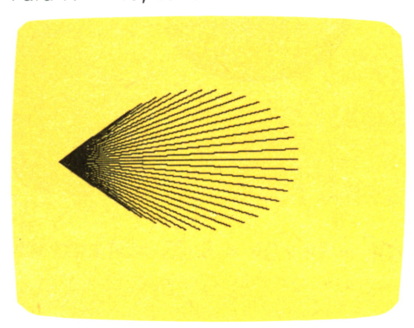


7. Rayos que parten de un punto situado a la izquierda de una circunferencia

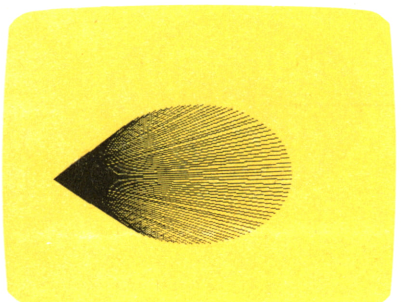
Localizando el centro emisor de los rayos del programa anterior en el pixel (27, 95) (para que quede suficiente espacio a su derecha) y aumentando X en 100 unidades en la instrucción **LINE - STEP,** se obtiene un conjunto de rayos que parten de un punto situado a la izquierda del centro de la circunferencia de radio 70, y centro el pixel (127, 95).

```
 10  INPUT "NUMERO DE RAYOS"; N
 20  SCREEN 2
 30  FOR I = 1 TO N
 40  PSET (27, 95)
 50  X = 70 * COS (6.2832/N * I)
 60  Y = 70 * SIN (6.2832/N * I)
 70 LINE - STEP (100 + X, Y)
 80  NEXT I
200  GOTO
```

Para N = 40, tenemos:



Para N = 100, obtenemos:



3. Representación gráfica de funciones

1. Planteamiento del problema

- Para dibujar la gráfica de una función debemos tener en cuenta dos cuestiones previas:

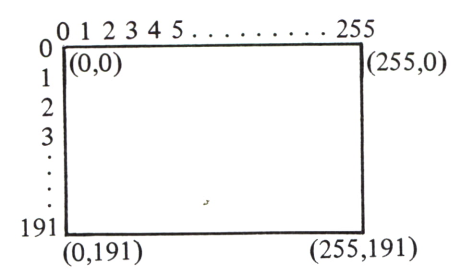
  — ¿Dónde situar el centro de los ejes?
  — ¿Qué escala hay que elegir en cada eje?

  Con el fin de contestar a estas preguntas estudiaremos una función sencilla: F (x) = x^2, y a partir de ella intentaremos sacar conclusiones generales.

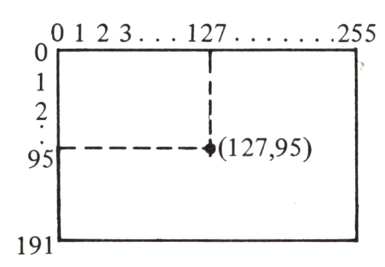
  Se supone que la pantalla del microordenador, en el modo gráfico, tiene 256 líneas verticales (de 0 a 255) y 192 líneas horizontales (de 0 a 191); luego en la pantalla se pueden localizar 256 × 192 = 49152 puntos.

  A cada punto de la pantalla se le llamará pixel.

  También se supone que en la pantalla en modo gráfico cada pixel está referido a un sistema de coordenadas, cuyo origen está localizado en el vértice superior izquierda (columna 0, fila 0). A los demás vértices les corresponden las coordenadas indicadas en la figura.



El centro de la pantalla corresponde entonces al pixel (127, 95).



- Las instrucciones específicas que se van a utilizar se resumen en esta tabla:

Instrucción	*Función*
SCREEN 2	Pasa a modo gráfico.
PSET (X, Y)	Imprime un punto en el pixel (X, Y).
LINE (A, B) – (X, Y)	Dibuja un segmento desde el pixel (A, B) al pixel (X, Y).
n GOTO n	Mantiene la imagen en la pantalla. (En los microordenadores MSX, presionar las teclas [CTRL] y [STOP] para salir del bucle sin fin.)

2. Centrado de ejes

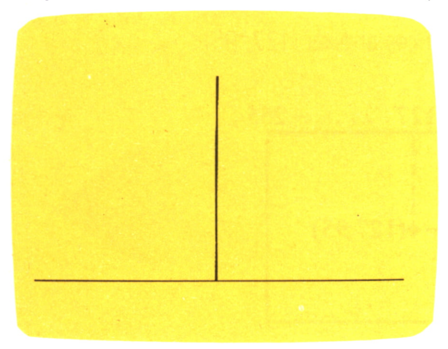
La gráfica de la función **F (x)** = x^2 es una parábola cuyo vértice es el punto (0, 0), y ocupa el primero y el segundo cuadrante. Luego los ejes deben estar centrados en el pixel (127, 191) que se encuentra en el centro de la línea inferior de la pantalla.

La subrutina que dibuja los ejes es:

```
1000  REM CENTRADO DE EJES
1010  SCREEN 2
1020  LINE (0, 191) – (255, 191)
1030  LINE (127, 191) – (127, 0)
2000  GOTO 2000
```

La instrucción **1020** dibuja el eje de abscisas, y la instrucción **1030** dibuja el eje de ordenadas.

Al ejecutar esta subrutina se obtiene lo que muestra la pantalla.

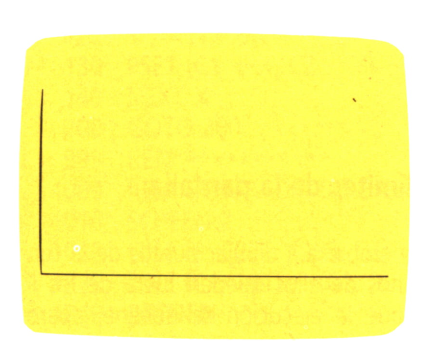


Si queremos que los ejes queden centrados en el pixel (C1, C2), basta generalizar la subrutina anterior cambiando 127 por C1 y 191 por C2 en las instrucciones **1030** y **1010,** respectivamente.

```
1000 REM CENTRADO DE EJES
1010 SCREEN 2
1020 LINE (0, C2) – 255, C2)
1030 LINE (C1, 191) – (C1, 0)
2000 GOTO 2000
```

Esta subrutina dibuja los ejes coordenados para cualquier centro (C1, C2), pero aquí solamente consideraremos los siguietes:

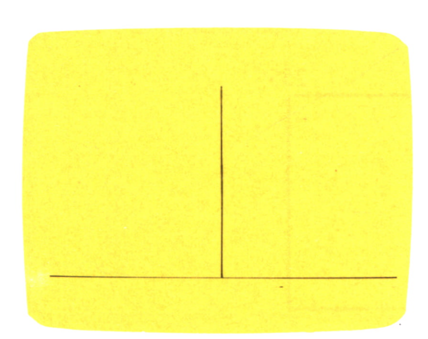
- **Centro (0, 191): determina el primer cuadrante**



Para conseguir los ejes con este centro hay que añadir estas instrucciones:

```
  20 READ C1, C2
9000 REM DATOS
9010 DATA 0, 191
```

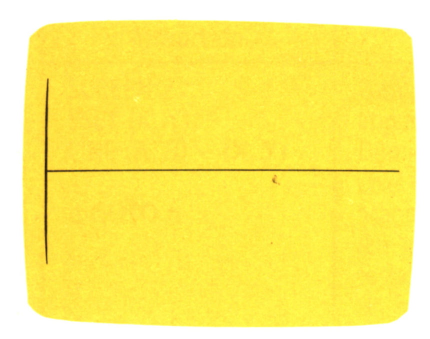
- **Centro (127, 191): determina el primero y el segundo cuadrante**



Para conseguir los ejes con este centro hay que añadir estas instrucciones:

```
  20 READ C1, C2
9000 REM DATOS
9010 DATA 127, 191
```

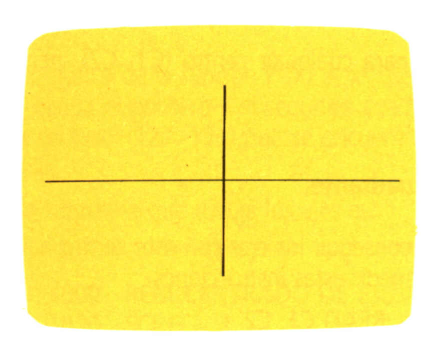
- **Centro (0, 95): determina el primero y el cuarto cuadrante:**



Para conseguir los ejes con este centro hay que añadir estas instrucciones:

```
  20 READ C1, C2
9000 REM DATOS
9010 DATA 0, 95
```

- **Centro (127, 95): determina los cuatro cuadrantes**

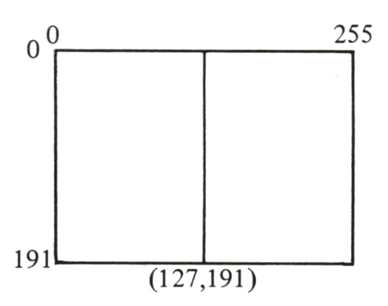


Para conseguir los ejes con este centro hay que añadir estas instrucciones:

```
  20 READ C1, C2
9000 REM DATOS
9010 DATA 127, 95
```

3. Localización de la gráfica en los límites de la pantalla

- Si una vez elegido el centro de los ejes, se empieza a dibujar puntos de la función F (X) = X^2, puede ocurrir que algunos de ellos queden fuera de los límites de la pantalla. Ocurre, entonces, que la ejecución se detiene, apareciendo un mensaje de error. Es necesario, por tanto, evitar esta situación eligiendo adecuadamente los valores externos de X a la izquierda y derecha de la pantalla, de tal manera que sus correspondientes valores F (X) no sobrepasen los valores 191 y 0 (límite inferior y superior de la pantalla).



Si el valor máximo que puede tomar F (X) es 191, ¿cuáles serán los valores extremos de X que hacen que F (X) no supere dicho valor?

Si $F(X) = X^2 = 191$ entonces $X = \pm \sqrt{191} = \pm 13.8203$.

Luego los valores de X estarán comprendidos entre −13 y 13:

$$-13 < X < 13$$

- Los valores extremos de X a la izquierda y a la derecha del origen de coordenadas los indicaremos por X1 y X2, respectivamente. En el ejemplo que venimos estudiando, X1 = −13 y X2 = 13. Fijados estos extremos, ya se puede dibujar la gráfica de la función **F (X)** = $\mathbf{X^2}$, mediante el siguiente programa:

```
  10  DEF FN F (X) = X ↑ 2
  20  READ C1, C2
  30  GOSUB 1000
  40  READ X1, X2
  99  REM *****
 100  REM DIBUJO
 110  FOR X = X1 TO X2
 120  Y = FN F (X)
 130  PSET (C1 + X, C2 − Y)
 140  NEXT X
 200  GOTO 200
 999  REM ***********
1000  REM CENTRADO DE EJES
1010  SCREEN 2
1020  LINE (0, C2) − (255, C2)
1030  LINE (C1, 191) − (C1, 0)
1040  RETURN
8999  REM ****
9000  REM DATOS
9010  DATA 127, 191
9020  DATA −13, 13
```

Este programa funciona así:

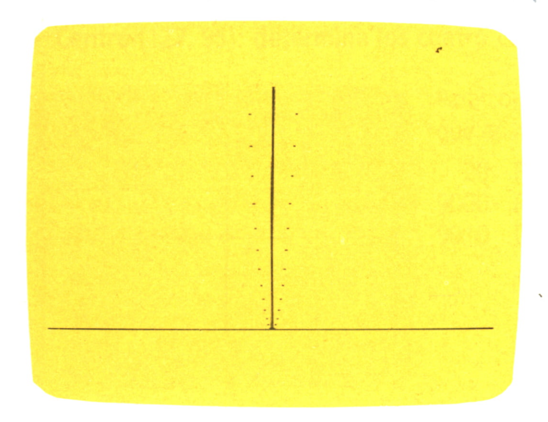
— La instrucción **10** define la función **FN F (X) = X ↑ 2**

— La instrucción **20** lee los valores del centro C1 y C2, y la instrucción **30** llama a la subrutina de «centrado de ejes» para que dibuje los ejes de coordenadas centrados en C1, C2.

— La instrucción **40** lee los valores extremos del eje de abscisas X1 y X2.

— El bloque **100 - 140** dibuja la función.

Al ejecutar el programa se obtiene la siguiente gráfica:



Se observa que la gráfica obtenida está formada por puntos bastante distantes, y que se ha desperdiciado gran parte de la pantalla, a izquierda y a derecha del eje de ordenadas. Esto se puede evitar eligiendo adecuadamente las escalas en ambos ejes.

4. Elección de escalas en los ejes

- Para determinar la escala en cada eje elegiremos las siguientes variables:

  X1: extremo inferior del eje de abscisas
  X2: extremo superior del eje de abscisas
  Y1: extremo inferior del eje de ordenadas
  Y2: extremo superior del eje de ordenadas
  DX: valor de la división en el eje X
  DY: valor de la división en el eje Y
  X3: abscisa del punto de la pantalla que se va a imprimir
  191 - Y3: ordenada del punto de la pantalla que se va a imprimir.

El programa que incluye la utilización de escalas es el siguiente:

```
  10 DEF FN F (X) = X ↑ 2
  20 READ C1, C2
  30 GOSUB 1000
  40 READ X1, X2
  50 READ Y1, Y2
  60 DX = (X2 - X1)/255
  70 DY = (Y2 - Y1)/191
  99 REM ******
 100 REM DIBUJO
 110 FOR X = X1 TO X2 STEP DX
 120 Y = FN F (X)
 130 X3 = INT ((X - X1) / DX + 0.5)
 140 Y3 = INT ((Y - Y1) / DY + 0.5)
 150 PSET (X3, 191 - Y3)
 160 NEXT X
 200 GOTO 200
 999 REM ***********
1000 REM CENTRADO DE EJES
1010 SCREEN 2
1020 LINE (0, C2) - (255, C2)
1030 LINE (C1, 191) - (C1, 0)
1040 RETURN
8999 REM *****
9000 REM DATOS
9010 DATA 127, 191
9020 DATA -13, 13
9030 DATA 0, 169
```

Veamos cómo funciona este programa:

— La instrucción **10** define la función **F (X) = X ↑ 2.**

— La instrucción **20** lee las coordenadas del centro (C1, C2).

— La **30** llama a la subrutina que dibuja los ejes.

— La **40** lee los valores extremos del eje de abscisas y la **50** los valores extremos del eje de ordenadas.

Como se ha visto, estos valores extremos no son arbitrarios sino que se calculan teniendo en cuenta los límites de la pantalla.

En este ejemplo, X1 y X2 se determinaron anteriormente:

X1 = −13 y X2 = 13

Los extremos de Y se calculan considerando que

$F(-13) = (-13)^2 = 169$ y $F(13) = 13^2 = 169$

y que se cumple $F(X) = X^2 \geqslant 0$

Luego,

Y1 = 0 e Y2 = 169

— Las instrucciones **60** y **70** calculan el valor de la división en los ejes de abscisas y ordenadas, respectivamente.

$$DX = \frac{X2 - X1}{255} = \frac{13 - (-13)}{255} = \frac{26}{255} = 0.10196078$$

$$DY = \frac{Y2 - Y1}{191} = \frac{169 - 0}{191} = 0.88481675$$

Luego en el eje X el salto de 1 pixel va a corresponder a un salto real de 0.10196078 (se ensancha casi 10 veces la gráfica), y el eje Y va a corresponder a un salto real de 0.88481675.

— La línea **110 FOR X = X1 TO X2 STEP DX** abre el bucle que dibuja la función. En total se van a imprimir 255 pixels, pues X varía desde X1 = −13 hasta X2 = 13 pero con paso DX = 0.10196078.

— La instrucción **120 Y = FN F (X)** calcula los valores de la función $\mathbf{F(X) = X^2}$ correspondientes a los valores comprendidos entre X1 y X2.

— Las instrucciones **130** y **140** calculan los valores X3 e Y3 en función de X e Y, respectivamente. Aquí se tienen en cuenta los extremos X1, X2 y las escalas elegidas DX y DY en los ejes coordenados.

Los valores de la variable X3 estarán comprendidos entre 0 y 255, y se calculan mediante la expresión:

$$X3 = INT\left(\frac{X - X1}{DX} + 0.5\right)$$

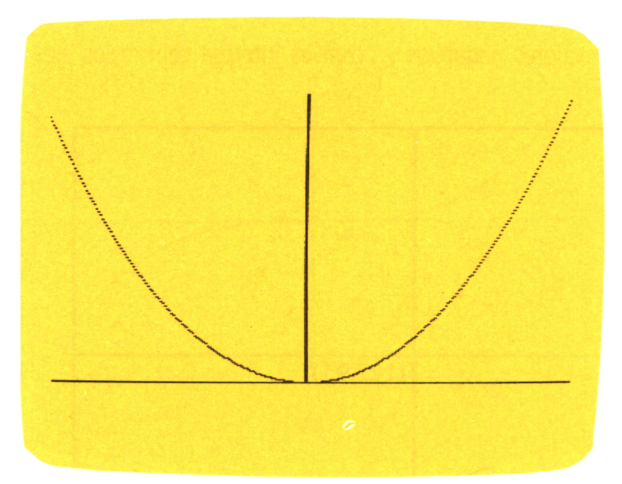
Los valores de Y3 estarán comprendidos entre 0 y 169, y se calculan mediante la expresión:

$$Y3 = INT\left(\frac{Y - Y1}{DY} + 0.5\right)$$

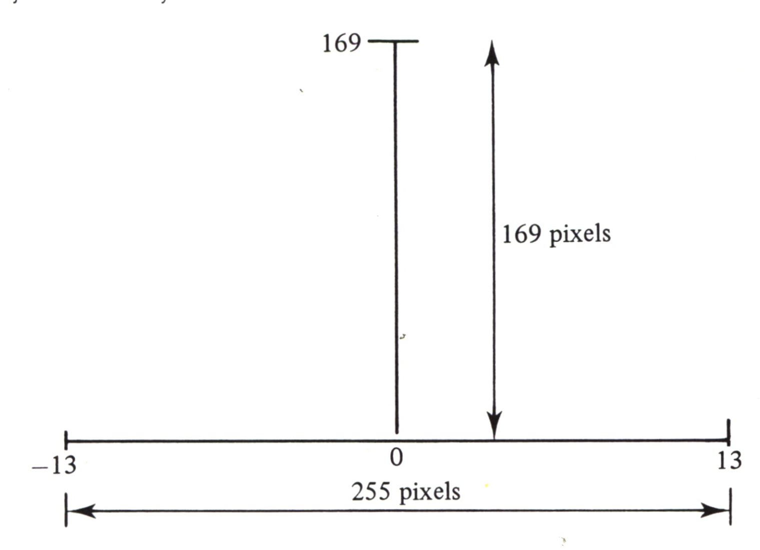
— La instrucción **150 PSET (X3, 191 − Y3)** imprime el pixel correspondiente.

— La línea **160 NEXT X** obliga a repetir el bucle.

Si se ejecuta el programa se obtiene lo que muestra la pantalla:



Ahora se aprecia mejor la gráfica de la función F (X) = X^2, aunque tiene el inconveniente de estar bastante deformada por haber elegido escalas diferentes en el eje de abscisas y en el de ordenadas.



La gráfica está ensanchada considerablemente, y este ensanchamiento viene dado por la relación entre ambas escalas DX y DY. Dicha relación es de 0.10196078 a 0.88481675.

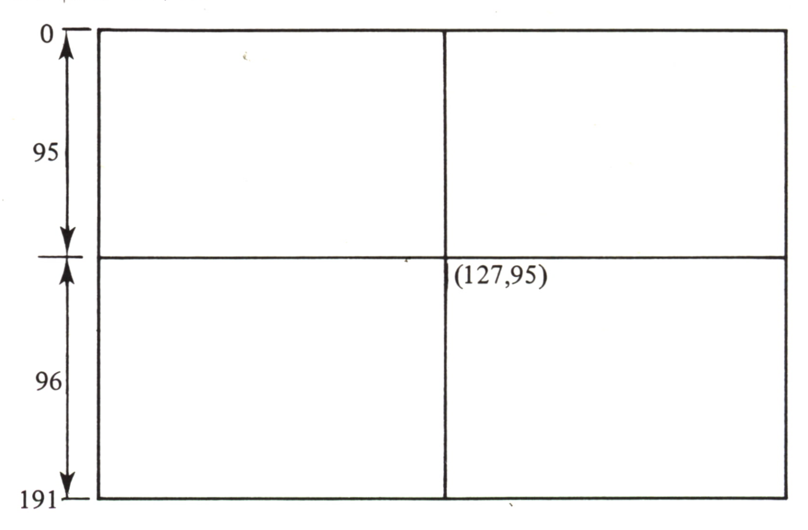
0.10196078 / 0.88481675 = 0.11523378

que impone un ensanchamiento de casi 10 veces.

5. Gráficas de algunas funciones

- $F(X) = X^3$

  Como F (X) puede tomar valores negativos y positivos interesa centrar los ejes en el pixel (127, 95).



Si hacemos $F(X) = X^3 = 95$, se obtiene $X = \sqrt{95} = 4.562903$.

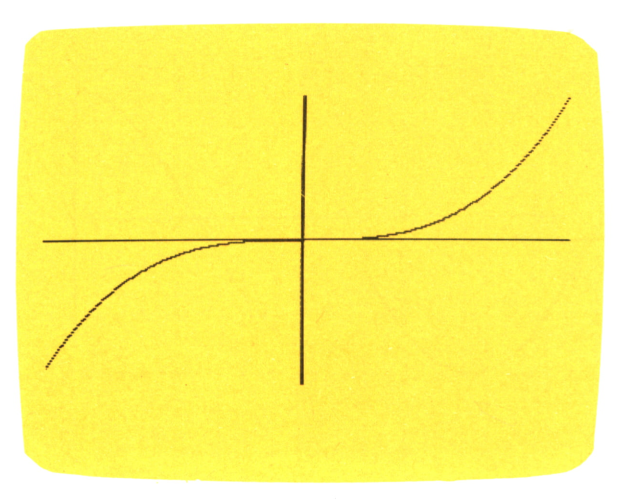
Elegiremos, entonces:

$X1 = -4 \qquad X2 = 4$
$Y1 = -64 \qquad Y2 = 64$

En el programa general hay que cambiar las instrucciones siguientes:

```
  10  DEF FN F (X) = X ↑ 3
9010  DATA 127, 95
9020  DATA -4, 4
9030  DATA -64, 64
```

Al ejecutar el programa se obtiene la siguiente pantalla:



- **F (X) = sen X**

La función **F (X) =sen X** es periódica, de período 2 π. Luego X variará entre X1 = 0 y X2 = 2 π = 6.2832. Como, además, el seno varía entre −1 y 1,

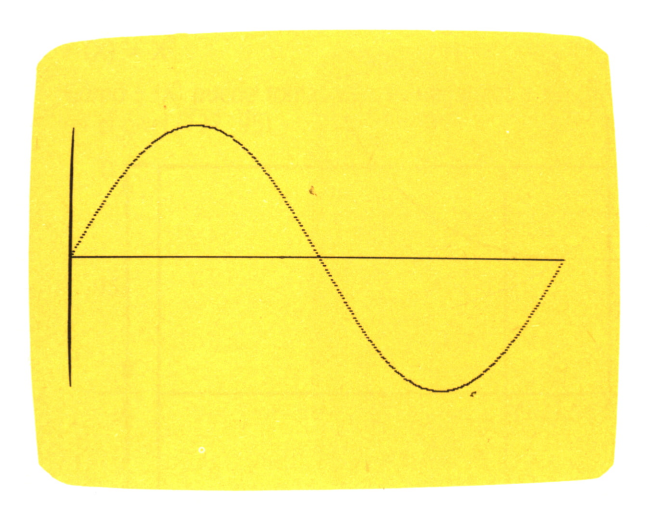
$$-1 \leqslant \text{sen } X \leqslant 1$$

los valores extremos de Y serán Y1 = −1 e Y2 = 1. Es evidente que el centro habrá que localizarlo en el pixel (0, 95).

Las instrucciones que hay que cambiar en el programa general son éstas:

```
  10 DEF FN F (X) = SIN (X)
9010 DATA 0, 95
9020 DATA 0, 6.2832
9030 DATA -1, 1
```

Al ejecutar el programa se obtiene la siguiente gráfica:



- F (X) = cos X

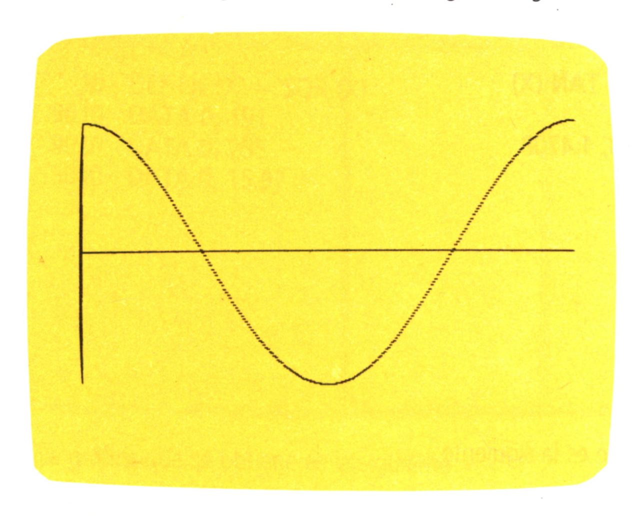
La función **F (X) = cos X** también es periódica, de período 2 π, luego X variará entre X1 = 0 y X2 = 2π = 6.2832. Como además se cumple −1 ≤ sen X ≤ 1, los valores extremos de Y serán los siguientes: Y1 = −1 e Y2 = 1.

El centro se situará en el pixel (0, 95).

Las instrucciones que hay que cambiar en el programa general son éstas:

```
  10 DEF FN F (X) = COS (X)
9010 DATA 0, 95
9020 DATA 0, 6.2832
9030 DATA -1, 1
```

Al ejecutar el programa se obtiene la siguiente gráfica:



- **F (X) = TAN X**

La función **F (X) = TAN X** también es periódica, pero de período y como es discontinua en

$$-\frac{\pi}{2}, \frac{\pi}{2}$$

elegiremos como valores extremos de X los siguientes:

$$X1 = -\frac{\pi}{2} + 0.1 = -1.4708$$

$$X2 = \frac{\pi}{2} - 0.1 = 1.4708$$

Y como

$$F(X1) = TAN\left(-\frac{\pi}{2} + 0.1\right) = -9.9666444$$

$$F(X2) = TAN\left(\frac{\pi}{2} - 0.1\right) = 9.9666444$$

hacemos

Y1 = −10

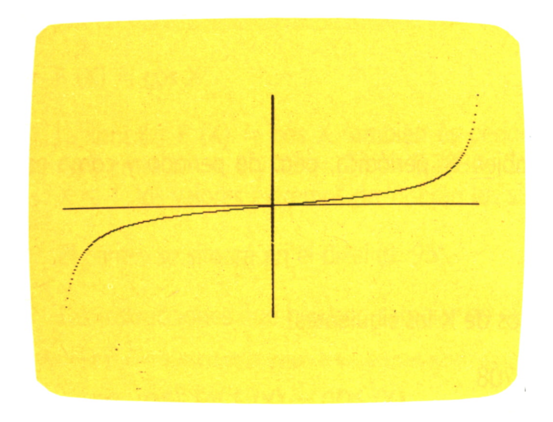
Y2 = 10

El centro estará en el pixel (127, 95).

Las instrucciones que hay que cambiar en el programa general son éstas:

```
  10  DEF FN F (X) = TAN (X)
9010  DATA 127, 95
9020  DATA -1.4708, 1.4708
9030  DATA -10, 10
```

La gráfica que se obtiene es la siguiente:



- F (X) = $\sqrt{X}$

La función **F (X)** = $\sqrt{X}$ da el valor positivo de la raíz cuadrada de X, y solamente definida para X $\geqslant$ 0. Se tiene entonces que si

X1 = 0
X2 = 255

la raíz cuadrada de estos valores es

$\sqrt{255}$ = 15.968719

Luego los valores extremos de Y son:

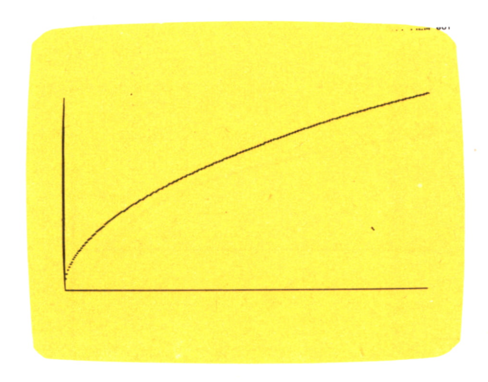
Y1 = 0
Y2 = 15.97

El centro estará en el pixel (0, 191).

Las instrucciones que hay que cambiar en el programa general son:

```
  10  DEF FN (X) = SQR (X)
9010  DATA 0, 191
9020  DATA 0, 255
9030  DATA 0, 15.97
```

La gráfica que se obtiene es la siguiente:



- F (X) = e^x

La función exponencial **F (X) = e^x** siempre es positiva y al ser $e^0 = 1$, en el eje de ordenadas debemos tomar un valor pequeño para que se pueda apreciar la gráfica de la función para valores de X negativos.

Si elegimos como valor extremo superior de Y

$$Y2 = 20$$

el correspondiente valor extremo de X a la derecha se obtiene así

$$e^x = 20 \Rightarrow \ln e^x = \ln 20 \Rightarrow x = 2.9957323$$

Esto nos lleva a elegir como valores extremos de X los siguientes:

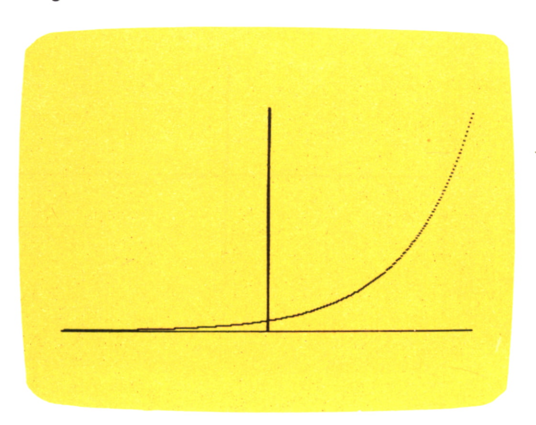
$$X1 = -3$$
$$X2 = 3$$

El centro será el pixel (127, 191).

Las instrucciones que hay que cambiar en el programa general son:

```
  10  DEF FN F (X) = EXP (X)
9010  DATA 127, 191
9020  DATA -3, 3
9030  DATA 0, 20
```

El resultado que se obtiene al ejecutar el programa es el que muestra la fotografía:



- F (X) = ln X

La función logaritmo natural o neperiano **F (X) = ln X** sólo está definida para valores de X positivos y al ser ln 1 = 0 el valor máximo de X no debe ser muy grande para que se pueda apreciar la parte negativa de la función.

Por ejemplo, si elegimos como valores extremos de X

X1 = 0.1
X2 = 20

los correspondientes valores extremos de Y se obtienen así:

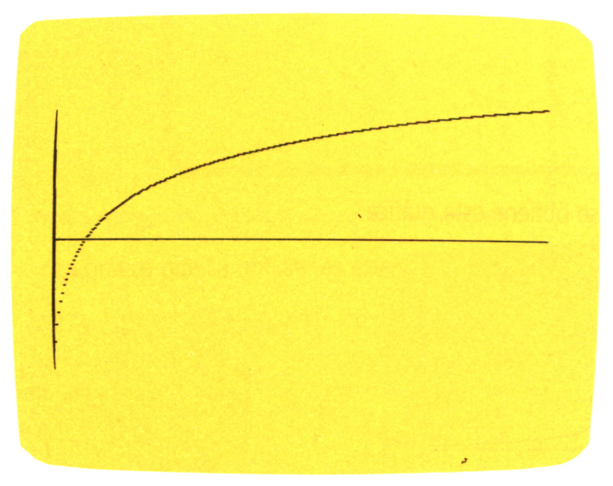
ln 0.1 = −2.3025851
ln 20 = 2.9957323

El centro será el pixel (0, 95).

Las instrucciones que hay que cambiar en el programa general son:

```
  10 DEF FN (X) = LOG (X)
9010 DATA 0, 95
9020 DATA 0.1, 20
9030 DATA -2.3, 3
```

La gráfica que se obtiene es la siguiente:



- $F(X) = \frac{1}{X}$

La función **F (X)** $= \frac{1}{X}$ es discontinua para X = 0, luego hay que evitar valores próximos al cero; además, X no debe ser ni muy pequeña ni muy grande, pues de lo contrario **F (X)** $= \frac{1}{X}$ se haría demasiado grande o demasiado pequeña, respectivamente. Así se obtendrá una buena representación.

Teniendo en cuenta las anteriores consideraciones, tomamos:

X1 = −5	X2 = 5
Y1 = −5	Y2 = 5

y como centro, el pixel (127, 95) pues X e Y van a ser ambos positivos.

Al dispararse la función para valores de X muy próximos al cero (discontinuidad en el cero) hay que añadir una instrucción como la siguiente:

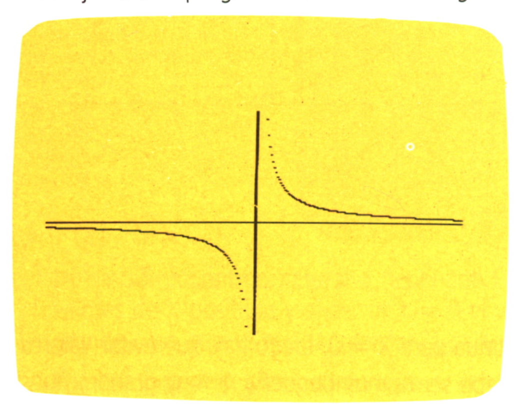
115 IF − 0.2 < = X AND X < = 0.2 THEN GOTO 160

que evita la división por cero.

Las instrucciones que hay que cambiar o añadir al programa general son:

```
  10  DEF FN (X) = 1/X
 115  IF −0.2 < = X AND X < = 0.2 THEN GOTO 160
9010  DATA 127, 95
9020  DATA −5, 5
9030  DATA −5, 5
```

Al ejecutar el programa se obtiene esta gráfica:



- **F (X)** = $\frac{\text{sen X}}{\text{X}}$

La función **F (X)** = $\frac{\text{sen X}}{\text{X}}$

es continua, pues cuando X se aproxima a cero (0), se tiene que **F (X)** = $= \frac{\text{sen X}}{\text{x}}$ se aproxima a 1. $\frac{\text{sen } 0.1}{0.1} \simeq 0.99833417$, y cuando X se hace

grande (en valor absoluto), $\frac{\text{sen } X}{X}$ tiende a cero, pues el sen X se mantiene entre −1 y 1. Así: $\frac{\text{sen } 15}{15} = 0.043352523$.

Teniendo en cuenta estas propiedades, elegimos como valores extremos, por ejemplo, los siguientes:

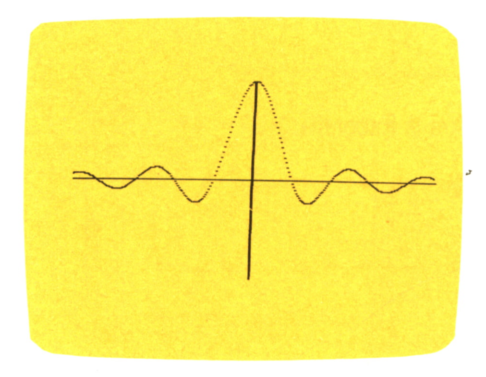
X1 = −15 X2 = 15
Y1 = −1 Y2 = 1

El centro será el pixel (127, 95), pues X e Y pueden ser positivos y negativos.

Las instrucciones que hay que cambiar en el programa general son:

```
  10  DEF FN F (X) = SIN (X) / X
9010  DATA 127, 95
9020  DATA -15, 15
9030  DATA -1, 1
```

La gráfica que se obtiene es ésta:



- F (X) = $\frac{\cos X}{X}$

Esta función es discontinua para X igual a cero, luego debemos añadir al programa general una instrucción como ésta:

```
115  IF -1 < = X AND X < = 1 THEN GOTO 160
```

dado que si X tomara un valor entre −1 y 1 la función se saldría de la pantalla.

Además, si X se hace grande (en valor absoluto) $\frac{\cos X}{X}$ tiende a cero, al mantenerse cos X entre −1 y 1. Así: $\frac{\cos 15}{15} = -0.050645861$.

Luego tomaremos por ejemplo los siguientes valores extremos:

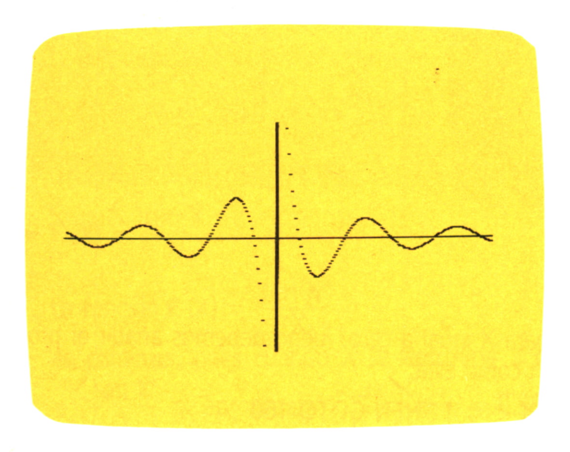
$X1 = -15 \qquad X2 = 15$
$Y1 = -1 \qquad Y2 = 1$

Como los valores de X e Y pueden ser positivos y negativos, el centro estará en el pixel (127, 95).

Las instrucciones que hay que añadir o cambiar en el programa general son:

```
  10 DEF FN F (X) = COS (X) / X
 115 IF -1 < = X AND X < = 1 THEN GOTO 160
9010 DATA 127, 95
9020 DATA -15, 15
9030 DATA -1, 1
```

La gráfica que se obtiene es la que muestra la fotografía:



- F (X) = sen X + cos X

La función **F (X) = sen X + cos X** es periódica, de período 2π, y al ser la suma del seno y del coseno, F (X) variará entre −1 −1 = −2 y 1 + 1 = 2.

Los valores extremos que debemos tomar son:

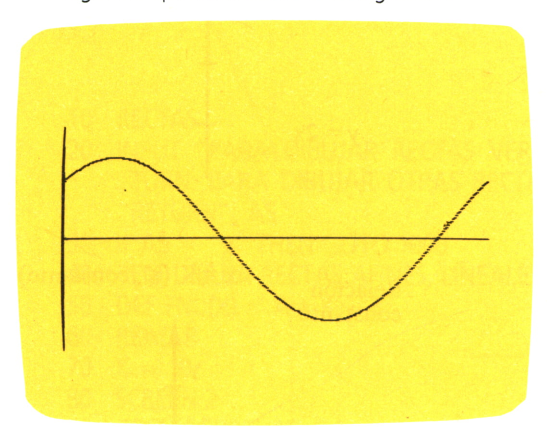
X1 = 0 X2 = 2π= 6.2832
Y1 = −2 Y2 = 2

Como X sólo toma valores positivos, el centro lo tomamos en el pixel (0, 95).

Las instrucciones que hay que cambiar en el programa general son:

```
  10 DEF FN (X) = SIN (X) + COS (X)
9010 DATA 0, 95
9020 DATA 0, 6.2832
9030 DATA −2, 2
```

La gráfica que se obtiene es là siguiente:

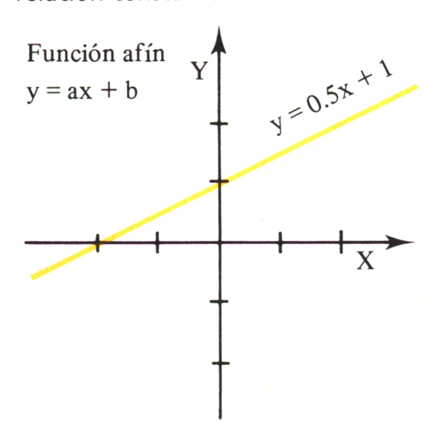


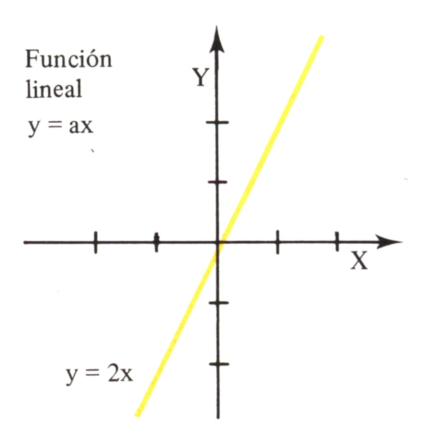
4. Rectas y parábolas. Resolución gráfica y algebraica de ecuaciones de primer y segundo grado

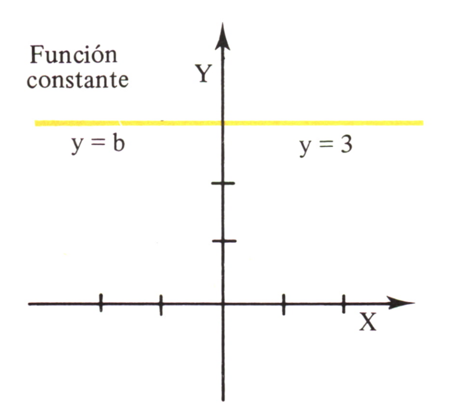
1. Representación gráfica de funciones afines, lineales y constantes, y de las relaciones constantes (familias de rectas)

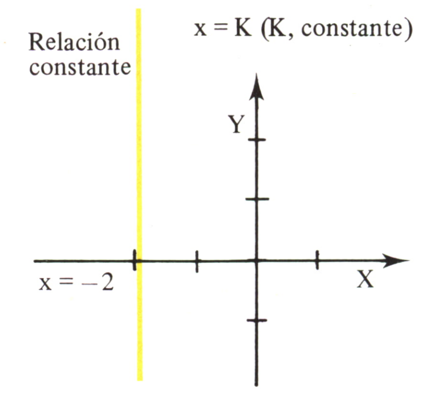
Presentación del problema

Transcribimos las fórmulas de la *función afín, lineal* y *constante*, así como de la *relación constante.*









Recordar que ***a*** es el coeficiente angular o ***pendiente*** de la recta ***afín*** y ***lineal,*** y es el parámetro que da la inclinación de la recta. El parámetro ***b*** es el ***término independiente,*** y determina el punto del eje de las ordenadas por el que pasa la recta. Si b = 0, la función afín se convierte en ***lineal,*** y en este caso la recta pasa por el origen de coordenadas.

Si a = 0, la función afín se convierte en ***constante,*** y en este caso la recta es paralela al eje de abscisas.

Si a > 0 y b = 0, la recta coincide con el eje de abscisas.

La exprexión x = k (k, constante) no es una función; es simplemente una relación. La gráfica correspondiente es una recta paralela al eje de ordenadas que corta el eje de abscisas en el punto k del mismo. Si k = 0, la recta coincide con el eje de ordenadas.

El programa

El siguiente programa dibuja gráficas de funciones afines, lineales y constantes, y de relaciones constantes.

También permite obtener en la pantalla familias de rectas, variando el coeficiente a y el término independiente b. El número máximo de rectas que pueden aparecen simultáneamente en la pantalla es 4.

Se ha puesto este tope porque con un número mayor, la pantalla puede aparecer confusa.

```
10 RECTAS
20 INPUT "PARA DIBUJAR RECTAS VERTICALES TECLEAR V Y RE-
     TURN. PARA DIBUJAR OTRAS RECTAS, PULSAR OTRA TECLA Y
     RETURN"; A$
30 IF A$ = "V" THEN GOTO 1000
40 REM DIBUJA RECTAS AFINES, LINEALES Y CONSTANTES
50 DEF FNI (X) = A * X + B
60 READ R
70 K = 10
80 SCREEN 2
90 OPEN "GRP:" AS # 1
100 GOSUB 2000
```

La instrucción **60 READ R** (y su correspondiente **DATA**) lee el dato R que da el número de rectas que se van a representar.

La instrucción **100** transfiere el control a la subrutina **2000** que dibuja los ejes de coordenadas.

```
110  FOR I = 1 TO R
120  READ A, B
130  IF K = 10 THEN PRESET(K, 6 * (I-1)) : PRINT # 1, "Y="; A;
     "X+"; B: K = 140: GOTO 150
140  IF K <> 10 THEN PRESET(K, 6 * (I-2)) : PRINT # 1, "Y="; A;
     "X+"; B : K = 10
150  GOSUB 3000
160  FOR T = 1 TO 1000: NEXT T
170  NEXT I
```

Las líneas **130** y **140** escriben en la parte superior de la pantalla de alta resolución la ecuación de la recta que se va a representar.

La instrucción **150** transfiere el control a la subrutina **3000** para dibujar la gráfica correspondiente.

La línea **160** introduce un bucle de retardo de unos dos segundos antes de dibujarse la siguiente recta.

```
180  PRESET(10, 183) : PRINT # 1, "Para seguir, CTRL y STOP.Y RUN"
190  GOTO 190
```

La línea **180** escribe en la parte inferior de la pantalla de alta resolución el texto que indica cómo salir del bucle sin fin de la línea **190.**

```
200  DATA
210  DATA
```

En **200 DATA** se pondrá el dato correspondiente al número R de rectas que se deseen representar y en **210 DATA,** el mismo número de pares de datos correspondientes a A y B.

```
1000  REM RECTAS VERTICALES
1010  INPUT "INTRODUCE ABSCISA X"; X
1020  SCREEN 2
1030  OPEN "GRP:" AS#1
1040  GOSUB 2000
1050  PRESET(120,0) : PRINT # 1, "X="; X
1060  LINE (129 + X/0.1, 20) – (129 + X/0.1, 180)
1070  PRESET(10, 180) : PRINT # 1, "Para seguir, CTRL y STOP.Y
       RUN"
1080  GOTO 1080
```

Este bloque de instrucciones dibuja rectas verticales, una vez introducido el valor de la abscisa por la cual pasará la recta.

En la línea **1060,** la variable X está dividida por 0.1 debido a que dos divisiones consecutivas del eje de abscisas están separadas 10 puntos.

```
2000  REM SUBRUTINA EJES
2010  REM EJE DE ABSCISAS
2020  LINE(0, 100) – (255, 100)
2030  FOR I = 7 TO 247 STEP 10
2040  PRESET(I, 97) : PRINT # 1, "|"
2050  NEXT I
2060  PRESET(138, 106) : PRINT # 1, "1"
2070  PRESET(247, 106) : PRINT # 1, "X"
2080  REM EJE DE ORDENADAS
2090  LINE (129, 20) – (129, 180)
2100  FOR I = 24 TO 170 STEP 10
2110  PRESET(127, I) : PRINT # 1, "_"
2120  NEXT I
2130  PRESET(120, 20) : PRINT # 1, "Y"
2140  PRESET(120, 86) : PRINT # 1, "1"
2150  RETURN
```

Esta subrutina dibuja los ejes coordenados con divisiones, señalando con X el eje de abscisas y con Y el de ordenadas, y poniendo un 1 en la primera división del eje X y del eje Y.

```
3000  REM DIBUJA LA GRAFICA
3010  IF A = 0 THEN X1 = –12 : X2 = 12 : GOTO 3060
3020  X1 = (–7–B)/A : X2 = (7 – B)/A
3030  IF A < 0 THEN C = X1 : X1 = X2 : X2 = C
3040  IF X1 < –12 THEN X1 = –12
3050  IF X2 > 12 THEN X2 = 12
```

Con este bloque de instrucciones se determinan los valores extremos X1 y X2 para evitar que los puntos de la gráfica queden fuera de la pantalla.

```
3060  FOR X = X1 TO X2 STEP 0.05
3070  Y = FNF(X)
3080  X3 = INT(X/0.1 + 0.5)
3090  Y3 = INT(Y/0.1 + 0.5)
3100  PSET(129 + X3, 100 – Y3)
3110  NEXT X
3120  RETURN
```

Las instrucciones **3060-3100** dibujan la gráfica. En las líneas **3080** y **3090,** las variables X e Y están divididas por 0,1 que establece la misma escala para los dos ejes. Se ha elegido este factor de escala debido a que dos divisiones consecutivas del eje de abscisas (y también del de ordenadas) están separadas 10 puntos.

La instrucción **3100** dibuja los puntos, refiriéndolos al pixel 129, 100 (centro de coordenadas).

Ejecución del programa

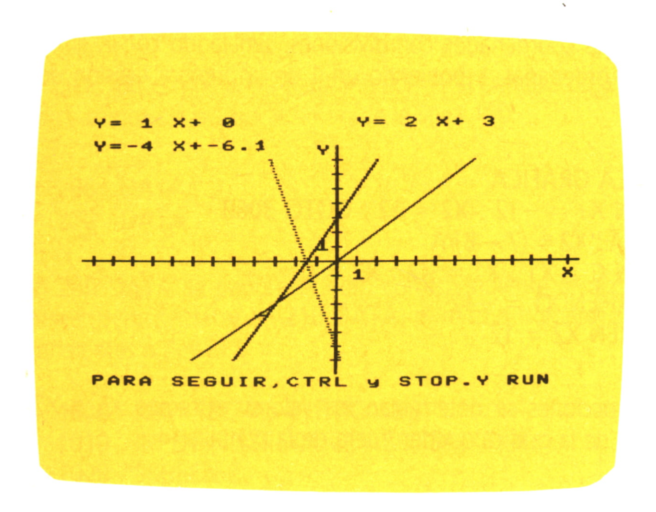
Antes de iniciar la ejecución del programa hay que poner en las instrucciones **200** y **210** los datos que se deseen.

A continuación presentamos algunos casos:

- Para representar gráficamente las funciones y = x, y = 2x + 3 e y = −4x − 6.1, las instrucciones **200** y **210** hay que escribirlas así:

  200 DATA 3
  210 DATA 1,0, 2,3, −4, −6.1

  Tecleando **RUN** y RETURN y pulsando a continuación cualquier tecla que no sea V se obtiene este resultado.

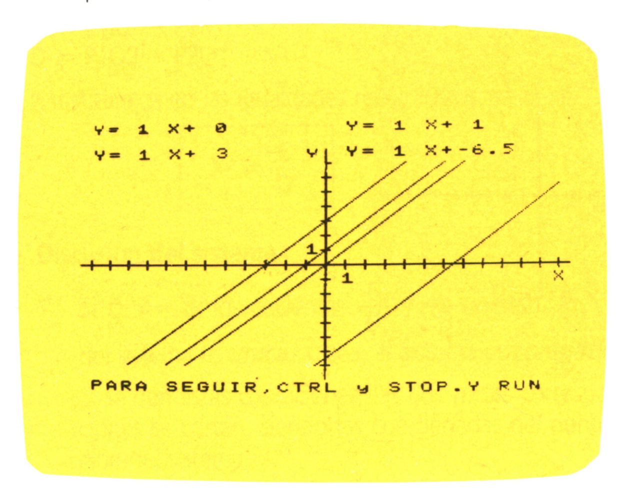


- Para obtener las gráficas de las funciones y = x, y = x + 1, y = x + 3 e y = x − − 6.5, las instrucciones **200** y **210** hay que escribirlas así:

  200 DATA 4
  210 DATA 1,0, 1,1, 1,3, 1, −6.5

  Tecleando **RUN** y RETURN y pulsando a continuación cualquier tecla que no

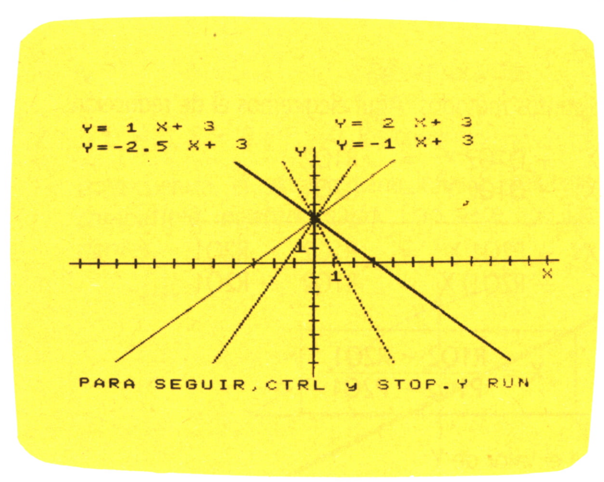
sea V se obtiene una familia de rectas con la misma pendiente (rectas paralelas de pendiente 1).



- Para representar gráficamente las funciones y = x + 3, y = 2x + 3, y = −2.5x + + 3 e y = −x + 3, las instrucciones **200** y **210** hay que escribirlas así:

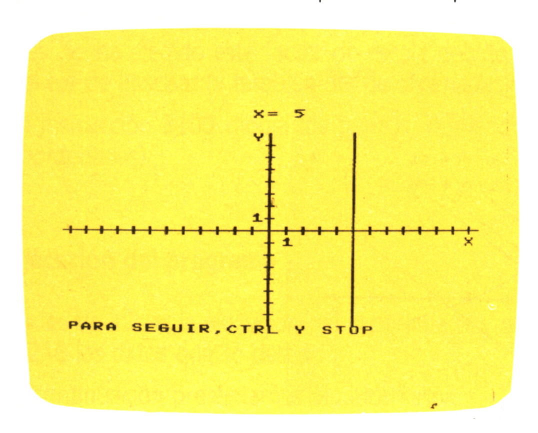
  200 DATA 4
  210 DATA 1,3, 2,3, −2.5,3, −1,3

  Tecleando **RUN** y RETURN y pulsando a continuación cualquier tecla que no sea V se obtiene una familia de rectas que cortan al eje Y por el punto 3.



- Si después de teclear **RUN** y RETURN se pulsa la tecla V, el programa continúa la ejecución en la instrucción, **1000 REM RECTAS VERTICALES** deteniéndose en la línea **1010 INPUT "INTRODUCE ABSCISA X"; X.**

Si se introduce el valor 5 para X en la pantalla se obtiene este resultado.



2. Resolución gráfica y algebraica de un sistema de primer grado con dos incógnitas

Planteamiento del problema

- ***Resolución del sistema***

El sistema de ecuaciones

$$\begin{cases} P1\,X + Q1\,Y = R1 \\ P2\,X + Q2\,Y = R2 \end{cases}$$

Se puede resolver por distintos métodos. Aquí elegiremos el de reducción:

$$\begin{array}{ll} (X\ Q2) & \left\{ \begin{array}{rrcr} P1Q2\,X & +\cancel{Q1Q2\,Y} & = & R1Q2 \\ -P2Q1\,X & -\cancel{Q1Q2\,Y} & = & -R2Q1 \end{array} \right. \\ (X\text{-}Q1) & \end{array}$$

$$\begin{array}{rrcrr} \hline P1Q2\,X & -\ P2Q1\,X & = & R1Q2 & -\ R2Q1 \\ (P1Q2 & -\ P2Q1)\,X & = & R1Q2 & -\ R2Q1 \end{array}$$

$$\boxed{X = \frac{R1Q2 - R2Q1}{P1Q2 - P2Q1}}$$

Análogamente se obtiene el valor de Y:

$$\boxed{Y = \frac{P1R2 - P2R1}{P1Q2 - P2Q1}}$$

Si hacemos

E = R1Q2 – R2Q1
F = P1R2 – P2R1
D = P1Q2 – P2Q1

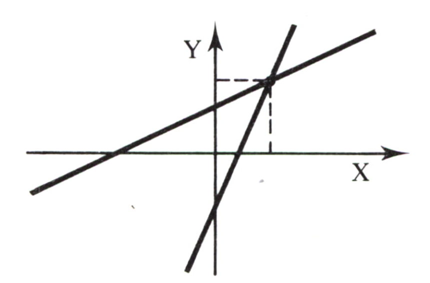
y sustituimos en las igualdades recuadradas, se tiene

$$\boxed{X = \frac{E}{D}} \qquad \boxed{Y = \frac{F}{D}}$$

- **Discusión del sistema**

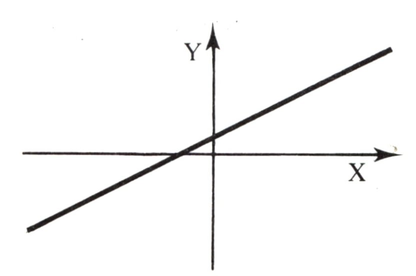
1. Si D ≠= 0, los cocientes $\frac{E}{D}$ y $\frac{F}{D}$ existen. En consecuencia, la solución del sistema es ***única***; o sea, el sistema es ***compatible determinado***.

   Si se dan estas condiciones, las dos rectas correspondientes a las dos ecuaciones se cortan, siendo las coordenadas del punto de intersección la solución del sistema.



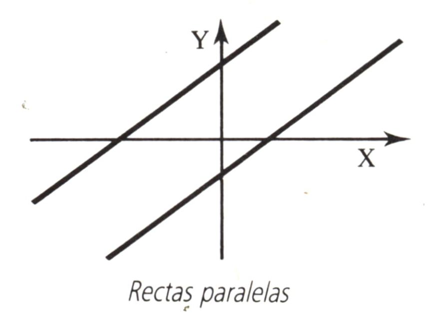
Rectas secantes

2. Si D = 0, E = 0 y F = 0, los cocientes $\frac{E}{D}$ y $\frac{F}{D}$ son indeterminados. En consecuencia, el sistema tiene infinitas soluciones; o sea, el sistema es ***compatible indeterminado***. Esto hace que las dos rectas sean ***coincidentes***.



Rectas coincidentes

3. Si D = 0, E = 0 o F = 0, el cociente $\frac{E}{D}$ o el cociente $\frac{F}{D}$ no existe. En consecuencia, el sistema no tiene solución; o sea, el sistema es ***incompatible.*** Se tiene, entonces, que las dos rectas son ***paralelas.***



Rectas paralelas

El programa

El siguiente programa resuelve gráficamente cualquier sistema de dos ecuaciones de primer grado con dos incógnitas, siempre que las rectas que correspondan a dichas ecuaciones caigan en los límites de la pantalla.

Una vez resuelto el sistema gráficamente, el programa analiza los valores de E, F y D para ***discutir*** el sistema, y si éste resulta ser compatible determinado, procede a calcular la solución. Si, en cambio, el sistema es compatible indeterminado o incompatible, da el mensaje correspondiente.

```
10  REM SISTEMA DE ECUACIONES
20  DEF FNF(X) = A * X + B
25  DEF FNR(X) = INT(100 * X + 0.5)/100
30  SCREEN 2
40  OPEN "GRP:" AS # 1
50  GOSUB 2000
60  REM RESOLUCION GRAFICA
70  FOR I = 1 TO 2
80  READ P, Q, R
85  IF I = 1 THEN PRESET(80, 0) : PRINT # 1, P; "X+"; Q; "Y=";
    R : GOTO 90
87  PRESET(80, 10) : PRINT # 1, P: "X+"; Q; "Y="; R
90  IF Q = 0 THEN X = R/P: GOTO 113
100 A = -P/Q : B = R/Q : GOSUB 3000
110 GOTO 115
113 GOSUB 4000
115 FOR T = 1 TO 1000 : NEXT T
120 NEXT I
130 DATA ..., ..., ..., ...
```

— La línea **50** transfiere el control a la subrutina **2000** que dibuja los ejes coordenados.

— Las líneas **70-120** dibujan las dos rectas. (La línea **100** transfiere el control precisamente a la subrutina **3000** que dibuja la gráfica de cada ecuación.)

— La línea **90** contempla el caso particular en que el coeficiente de Y es cero, en cuyo caso la gráfica de la ecuación es una recta vertical, o sea paralela al eje de ordenadas (la gráfica la dibuja la subrutina **4000**).

```
140  RESTORE
150  REM RESOLUCION ALGEBRAICA
160  FOR I = 1 TO 2
170  READ P, Q, R
180  IF I = 1 THEN P1 = P : Q1 = Q : R1 = R : GOTO 200
190  P2 = P : Q2 = Q : R2 = R
200  NEXT I
210  E = R1 * Q2 - R2 * Q1 : F = P1 * R2 - P2 * R1 : D =
     = P1 * Q2 - P2 * Q1
220  IF D = 0 AND(E = 0 AND F = 0) THEN GOTO 370
230  IF D = 0 AND(E<> 0 OR F <> 0) THEN GOTO 380
240  X = E/D : Y = F/D
```

— Con las instrucciones **140-200** se vuelven a leer los coeficientes de las dos ecuaciones.

— La **210** calcula el valor de E, F y D y las líneas **220** y **230** discuten el sistema.

— La línea **240** calcula X e Y (soluciones), en el caso de que el sistema sea compatible determinado.

```
250  IF X < -12 OR X > 12 OR Y > 7 OR Y < -7 THEN GOTO 360
255  IF Q1 = 0 OR Q2 = 0 THEN GOTO 305
260  H = 100 - Y/0.1
270  IF H < = 100 THEN S = 2 : GOTO 280
275  IF H > 100 THEN S = -3
280  FOR I = H TO 100 STEP S
290  PSET (129 + X/0.1, I)
300  NEXT I
```

— En la línea **250** se averigua si las rectas se cortan en los límites de la pantalla. En caso afirmativo, el bloque de instrucciones **260-300** traza una línea pun-

teada desde el punto de corte hasta el eje de abscisas; así se visualiza el valor de la incógnita X.

Si una de las rectas es vertical (hecho que sucede si Q1 = 0 o Q2 = 0) no se dibuja la línea punteada, transfiriéndose el control a la línea **305.**

```
305  IF P1 = 0 OR P2 = 0 THEN GOTO 360
310  H = 129 + X/0.1
320  IF H < = 129 THEN S = 3 : GOTO 340
325  IF H > 129 THEN S = −3
330  FOR I = H TO 129 STEP S
340  PSET (I, 100 − Y/0.1)
350  NEXT I
```

Estas instrucciones trazan una línea punteada desde el punto de corte hasta el eje de ordenadas; así se visualiza el valor de la incógnita Y.

Si una de las rectas es horizontal (hecho que sucede si P1 = 0 o P2 = 0 no se dibuja la línea punteada, transfiriéndose el control a la línea **360.**

```
 360  PRESET(0, 183) : PRINT #, "SIST. COMPATIBLE DETERMI-
      NADO": GOTO 390
 370  PRESET (0, 183) : PRINT # 1, "S. INDETERM. (RECT. COINCI-
      DENTES)" : GOTO 390
 380  PRESET(0, 183) : PRINT # 1, "S. INCOMP. (RECT. PARALELAS)"
 390  GOTO 390
 400  END
2000  REM SUBRUTINA EJES
2010  REM EJE DE ABSCISAS
2020  LINE (0, 100) − (255, 100)
2030  FOR I = 7 TO 247 STEP 10
2040  PRESET (I, 97) : PRINT # 1, "|"
2050  NEXT I
2060  PRESET(138, 106) : PRINT # 1, "1"
2070  PRESET(247, 106) : PRINT # 1, "X"
2080  REM EJE DE ORDENADAS
2090  LINE (129, 20) − (129, 180)
2100  FOR I = 24 TO 170 STEP 10
2110  PRESET (127, I) : PRINT # 1, "_"
2120  NEXT I
2130  PRESET(120, 20) : PRINT # 1, "Y"
2140  PRESET(120, 86) : PRINT # 1, "1"
2150  RETURN
```

Esta subrutina dibuja los ejes coordenados con divisiones, señalando con X el eje de abscisas y con Y el de ordenadas.

```
3000  REM DIBUJA LA GRAFICA
3010  IF A = 0 THEN X1 = −12 : X2 = 12 : GOTO 3060
3020  X1 = (−7 − B)/A: X2 = (7 − B)/A
3030  IF A < 0 THEN C = X1 : X1 = X2 : X2 = C
3040  IF X1 < −12 THEN X1 = −12
3050  IF X2 > 12 THEN X2 = 12
3060  FOR X = X1 TO X2 STEP 0.1
3070  Y = FNF(X)
3080  X3 = INT (X/0.1 + 0.5)
3090  Y3 = INT (Y/0.1 + 0.5)
3100  PSET(129 + X3, 100 + Y3)
3110  NEXT X
3120  RETURN

4000  REM RECTAS VERTICALES
4010  LINE(129 + X/0.1, 20) − (129 + X/0.1, 180)
4020  RETURN
```

Ejecuciones del programa

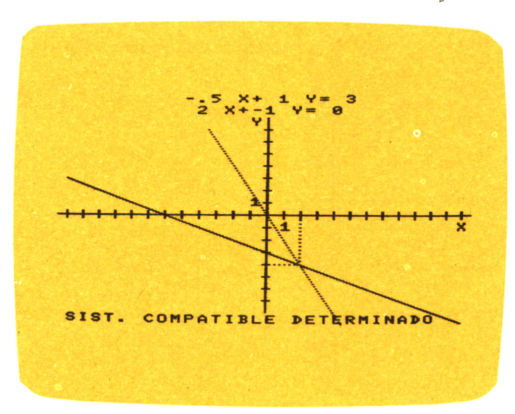
- Para resolver el sistema

$$\begin{cases} -0.5x + y = 3 \\ 2x - y = 0 \end{cases}$$

la instrucción **130** habrá que describirla así:

```
130  DATA −0.5, 1, 3, 2, −1, 0
```

Al ejecutar el programa con esta instrucción se obtiene este resultado.



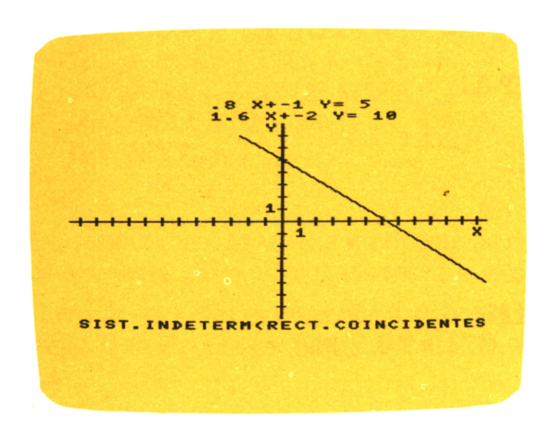
- Para resolver el sistema

$$\begin{cases} 0.8x - y = 5 \\ 1.6x - 2y = 10 \end{cases}$$

la instrucción **130** habrá que escribirla así:

130 DATA 0.8, –1, 5, 1.6, –2, 10

Al ejecutar el programa se obtiene este resultado.



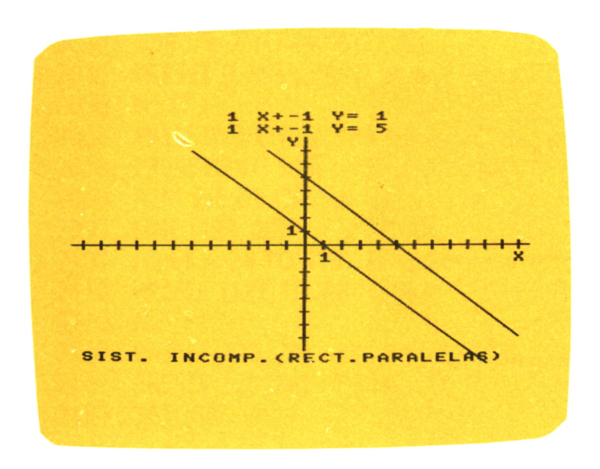
- Para resolver el sistema

$$\begin{cases} x - y = 1 \\ x - y = 5 \end{cases}$$

la instrución **130** habrá que escribirla así:

130 DATA 1, –1, 1, 1, –1, 5

Al ejecutar el programa se obtiene este resultado.



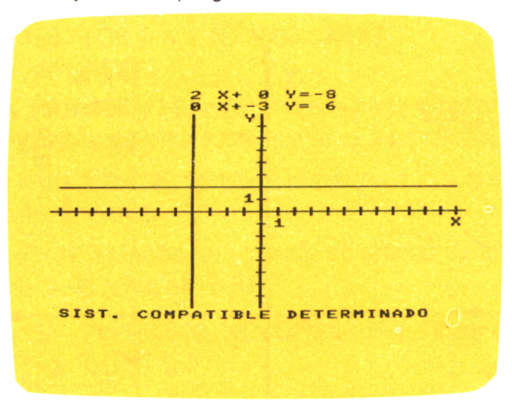
- Para resolver el sistema

$$\begin{cases} 2x + 0y = -8 \\ 0x - 3y = 6 \end{cases}$$

la instrucción **130** hay que escribirla así:

130 DATA 2, 0, –8, 0, –3, 6

Al ejecutar el programa se obtiene este resultado



3. Representación gráfica de la función cuadrática $f(x) = (mx + n)^2 + p$. (Familia de parábolas)

Presentación del problema

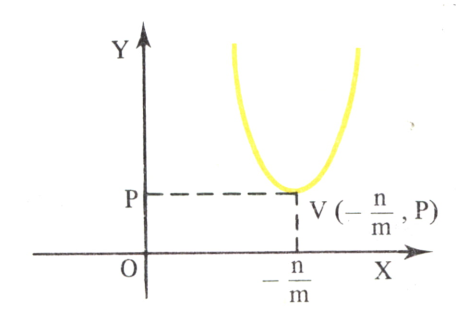
La función cuadrática $f(x) = ax^2 + bx + c$ se puede expresar en la forma

$$f(x) = (mx + n)^2 + p$$

cuya gráfica es una parábola.

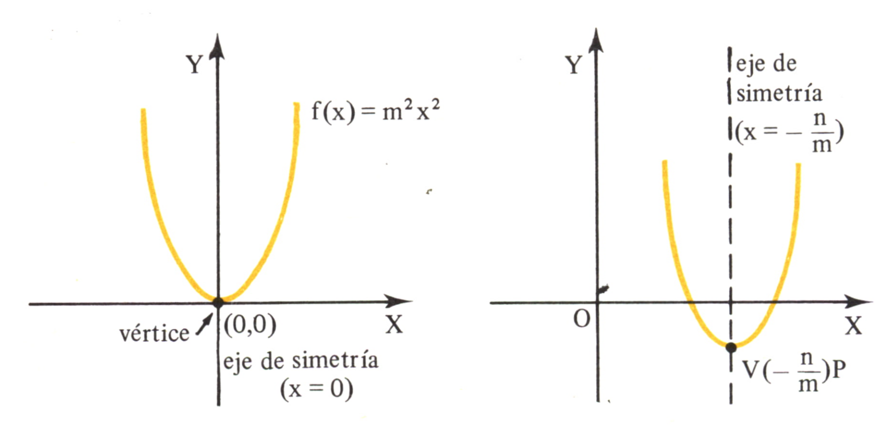
Los parámetros m, n y p determinan la forma de la parábola (si es más abierta o cerrada) y su posición.

El punto característico de la parábola es su vértice V, cuyas coordenadas son $(-\frac{n}{m}\ p)$.



La recta vertical que pasa por el vértice de la parábola es su eje de simetría y su ecuación es $x = -\frac{n}{m}$.

Si **n** = **0** y **p** = **0** la ecuación de la parábola es $f(x) = (mx)^2 = m^2 - x^2$ cuyo vértice coincide con el origen de coordenadas: $\left(-\frac{0}{p}, 0\right) = (0,0)$. En consecuencia, el eje de simetría de esta parábola es el eje de ordenadas.



La recta vertical que pasa por el vértice de la parábola es su eje de simetría, y su ecuación es $x = -\frac{n}{m}$.

El programa

El siguiente programa dibuja gráficos de funciones cuadráticas. También permite obtener en la pantalla familia de parábolas, variando los parámetros m, n y p. El número máximo de parábolas que pueden aparecer en la pantalla es 4. Se ha puesto este tope porque con un número mayor la pantalla puede aparecer confusa.

```
10  REM PARABOLAS
20  DEF FNF(X) = (M * X + N) ∧ 2 + P
30  READ G
50  SCREEN 2
60  OPEN "GRP:" AS # 1
70  GOSUB 2000
```

La instrucción **30 READ G** (y su correspondiente **DATA**) lee el dato G que da el número de parábolas que se van a representar.

La instrucción **70** transfiere el control a la subrutina **2000** que dibuja los ejes coordenados.

```
80  FOR I = 1 TO G
90  READ M, N, P
100 IF I – 1 THEN PRESET (10,0) : PRINT # 1, "Y = ("; M; "X+"; N;")
    > 2 +"; P : GOTO 120
110 IF I = 2 THEN PRESET (10,0) : PRINT # 1, "Y = ("; M; "X+"; N;")
    > 2 +"; P : GOTO 120
115 PRESET (10,20) : PRINT # 1, "Y = ("; M; "X + "; N; ")∧2 + " ; P
120 GOSUB 3000
130 FOR T = 1 TO 1000 : NEXT T
140 NEXT I
```

Las líneas **100** y **110** escriben en la parte superior de la pantalla de alta resolución la ecuación de la parábola que se va a representar.

La instrucción **120** transfiere el control a la subrutina **3000** para dibujar la gráfica correspondiente.

La línea **130** introduce un bucle de retardo de unos dos segundos antes de dibujarse la siguiente parábola:

```
150 PRESET (10, 183) : PRINT # 1, "Para salir, CTRL y STOP"
160 GOTO 160
```

La línea **150** escribe en la parte inferior de la pantalla de alta resolución el texto que indica cómo salir del bucle sin fin de la línea **160.**

```
170 DATA
180 DATA
190 END
```

En **170 DATA** se pondrá el dato correspondiente al número de parábolas que se desean representar (G) y en **180 DATA,** el mismo número de ternas de datos correspondientes a M, N y P.

```
2000 REM SUBRUTINA EJES
2010 REM EJE DE ABSCISAS
2020 LINE (0, 110) – (255, 110)
2030 FOR I = 7 TO 247 STEP 10
2040 PRESET(I, 107) : PRINT # 1, "|"
2050 NEXT I
2060 PRESET(138, 116) : PRINT # 1, "1"
2070 PRESET(247, 116) : PRINT # 1, "X"
2080 REM EJE DE ORDENADAS
2090 LINE (129, 30) – (129, 182)
2100 FOR I = 34 TO 180 STEP 10
2110 PRESET (127, I) : PRINT # 1, "_"
2120 NEXT I
2130 PRESET(120, 30) : PRINT # 1, "Y"
2140 PRESET(120, 96) : PRINT # 1, "1"
2150 RETURN
```

Esta subrutina dibuja los ejes coordenados con divisiones, señalando con X el eje de abscisas y con Y el eje de coordenadas, y poniendo un 1 en la primera división del eje X y del eje Y.

```
3000 REM DIBUJA LA GRAFICA
3010 E = –N/M : IF E < –12 OR E > 12 THEN GOTO 3160
3015 IF 7 – P < 0 THEN PRESET (10,30) : PRINT # 1, "Si p > 7 no se
     puede repres." : GOTO 3160
3020 X1 = ((–1) * SQR (7 – P) – N)/M : X2 = (SQR(7 – P) – N)/M
3030 IF X1 > X2 THEN C = X1 : X1 = X2 : X2 = C
3040 IF X1 < –12 THEN X1 = –12
3050 IF X2 > 12 THEN X2 = 12
3070 FOR X = X1 TO X2 STEP 0.05
3080 Y = FNF(X)
3090 X3 = INT(X/0.1 + 0.5) : Y3 = INT (Y/0.1 + 0.5)
3095 IF 110 – Y 3 > 180 THEN GOTO 3110
3100 PSET(129 + X3, 110 – Y3)
3110 NEXT X
3120 REM EJE DE SIMETRIA
3130 FOR J = 30 TO 182
3135 D = J/6 : IF D = INT(D) THEN GOTO 3150
3140 PSET (129 + E/0.1, J)
3150 NEXT J
3160 RETURN
```

La línea **3010** calcula la abscisa del vértice de la parábola y, además, averigua si esta abscisa queda fuera de la pantalla, a la izquierda ($E < -12$) o a la derecha ($E > 12$). Si esto ocurre se decide no dibujar la gráfica, transfiriendo el control a la línea **3160**. La línea **3015** evita que la ejecución se bloquee en la línea **3020** cuando $7 - P < 0$, ya que la raíz cuadrada de un número negativo no existe en el conjunto de los números reales.

El bloque de instrucciones **3020-3050** determina el intervalo [X1, X2].

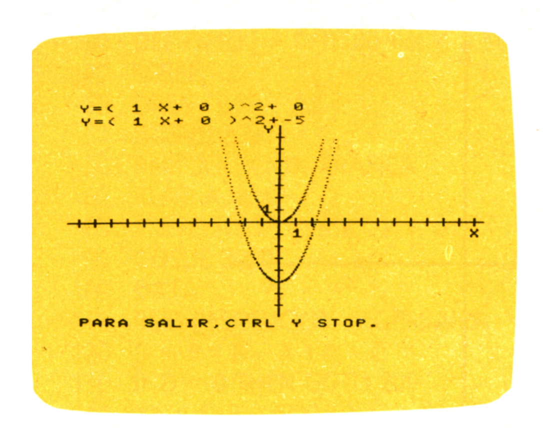
El bloque **3070-3110** representa la parábola.
Las instrucciones **3130-3150** dibujan el eje de simetría. La línea **3095** hace que no se representen en la pantalla los puntos de la parábola cuyas ordenadas sean menores que -7, respecto al origen de coordenadas elegido: (129, 110).

Ejecución del programa

- Si en las instrucciones **170** y **180** se ponen los datos que figuran a continuación:

```
170 DATA 2
180 DATA 1, 0, 0, 1, 0, –5
```

se obtiene este resultado:



- Las ecuaciones siguientes corresponden a 3 parábolas pertenecientes a la misma familia (el parámetro que varía es n).

$$y = (0.8x + 0)^2 - 2 = (0.8x)^2 - 2 = 0.64x^2 - 2$$
$$y = (0.8x + 1)^2 - 2$$
$$y = (0.8x - 4.3)^2 - 2$$

Para dibujar estas parábolas, las instrucciones **170** y **180** deberán escribirse así:

```
170 DATA 3
180 DATA 0.8, 0, –2, 0.8, 1, –2, 0.8, –4.3, –2
```

- Las ecuaciones siguientes corresponden a 3 parábolas de la misma familia (el parámetro que varía es m):

$$y = (x + 4)^2 - 5$$
$$y = (2x + 4)^2 - 5$$
$$y = (0.5x + 4)^2 - 5$$

Para dibujar estas parábolas las instrucciones **170** y **180** deberán escribirse así:

```
170 DATA 3
180 DATA 1, 4, –5, 2, 4, –5, 0.5, 4, –5
```

4. Resolución gráfica y algebraica de la ecuación de segundo grado

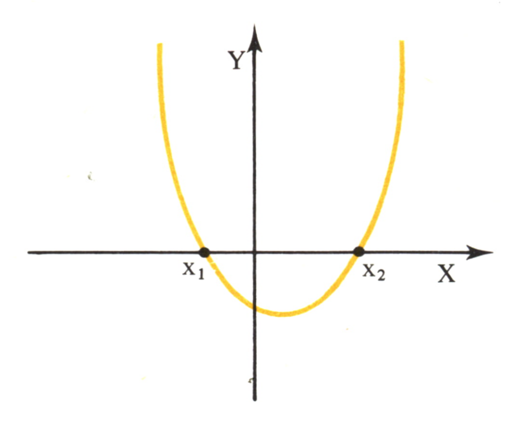
Gráficamente, las soluciones x_1 y x_2 de la ecuación de 2.° grado

$$ax^2 + bx + c = 0$$

vienen dadas por los puntos de corte de la parábola

$$y = ax^2 + bx + c$$

con el eje de abscisas:



Es evidente que si la parábola no corta al eje de abscisas la ecuación $\mathbf{ax^2 + bx + c = 0}$ no tiene soluciones reales.

- Algebraicamente, las soluciones de la ecuación $\mathbf{ax^2 + bx + c = 0}$ se obtienen mediante las conocidas fórmulas

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{y} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

  El radicando $b^2 - 4ac$ que aparece en estas fórmulas se llama ***discriminante,*** D. Este valor determina el tipo de solución de la ecuación de 2.° grado:

  — Si $D > 0$, se tiene que $\sqrt{D} > 0$ y, en consecuencia, las soluciones x_1 y x_2 serán ***reales*** y ***distintas.***

  — Si $D = 0$ se tiene que $\sqrt{D} = 0$ y, en consecuencia, las soluciones x_1 y x_2 serán ***reales*** y ***coincidentes;*** es decir, la ecuación tendrá una sola solución real.

  — Si $D < 0$, se tiene que $\sqrt{D}$ no es un valor real (es imaginario) y, en consecuencia, la ecuación ***no tiene soluciones reales.***

El programa

El siguiente programa resuelve gráficamente cualquier ecuación $\mathbf{ax^2 + bx + c = 0}$ siempre que tenga soluciones reales y siempre que la parábola de la función correspondiente $\mathbf{y = ax^2 + bx + c}$ entre en los límites de la pantalla.

Una vez resuelta la ecuación gráficamente, el programa la ***discute,*** y si tiene soluciones reales procede a calcularlas. Si, en cambio, las soluciones no son reales, da el mensaje correspondiente.

```
10  REM ECUACION DE SEGUNDO GRADO
20  DEF FNF(X) = (M * X + N) ∧ 2 + P
30  DEF FNR(X) = INT(100 * X + 0.5)/100
40  SCREEN 2
50  OPEN "GRP:" AS # 1
60  GOSUB 2000
70  REM SOLUCION GRAFICA
80  READ A, B, C
85  PRESET(10,0) : PRINT # 1, A; "X ∧ 2+"; B; "X+"; C; "=0"
90  GOSUB 3000
100 DATA
110 REM RESOLUCION ALGEBRAICA
120 D = B * B - 4 * A * C
130 IF D < 0 THEN GOTO 200
140 X1 = (-B + SQR(D))/(2 * A) : X2 = (-B - SQR(D))/(2 * A)
150 IF X1 > X2 THEN L = X1 : X1 = X2 : X2 = L
160 IF X1 < -12 OR X2 > 12 THEN GOTO 190
170 PRESET(129 + X1/0.1 - 14, 106) : PRINT # 1, "X1"
180 PRESET(129 + X2/0.1 + 5, 106) : PRINT # 1, "X2"
190 PRESET(10,10) : PRINT # 1, "X1="; FNR(X1); "X2="; FNR(X2) :
    GOTO 210
200 PRESET(10,10) : PRINT # 1, "NO TIENE SOLUCIONES REALES"
210 PRESET(10, 183) : PRINT # 1, "Para salir, CTRL y STOP."
220 GOTO 220
230 END
```

Este bloque de instrucciones constituye el programa principal.

— La instrucción **90** transfiere el control a la subrutina **3000** que dibuja la gráfica de la función $\mathbf{y = ax^2 + bx + c}$.

— La instrucción **120** calcula el discriminante D y la **130** averigua si la ecuación no tiene soluciones reales.

— La línea **140** calcula las soluciones.

— Las líneas **170** y **180** escriben en la pantalla de alta resolución los símbolos X1 y X2 junto a los puntos de corte de la parábola con el eje de abscisas, siempre que ello sea posible (fijarse en la línea **150**).

— La línea **190** escribe en la parte superior de la pantalla las soluciones de la ecuación, siempre que sean reales, y la línea **200** escribe el mensaje NO TIENE SOLUCIONES REALES (fijarse en la línea **130**).

— La línea **210** escribe, en la parte inferior de la pantalla cómo proceder para salir del bucle sin fin de la línea **220.**

— La instrucción **230** da por terminado el programa.

```
2000 REM SUBRUTINA EJES
2010 REM EJE DE ABSCISAS
2020 LINE (0, 100) – (255, 100)
2030 FOR I = 7 TO 247 STEP 10
2040 PRESET(I, 97) : PRINT # 1, "|"
2050 NEXT I
2060 PRESET(138, 106) : PRINT # 1, "1"
2070 PRESET(247, 106) : PRINT # 1, "X"
2080 REM EJE DE ORDENADAS
2090 LINE (129, 20) – (129, 180)
2100 FOR I = 24 TO 170 STEP 10
2110 PRESET(127, I) : PRINT # 1, "_"
2120 NEXT I
2130 PRESET(120, 20) : PRINT # 1, "Y"
2140 PRESET(120, 86) : PRINT # 1, "1"
2150 RETURN
```

Esta subrutina dibuja los ejes coordenados con divisiones, señalando con X el eje de abscisas y con Y el de ordenadas, y poniendo un 1 en la primera división del eje X y del eje Y.

```
3000 REM DIBUJA LA GRAFICA
3010 IF A < 0 THEN A1 = –A : B1 = –B : C1 = –C : GOTO 3030
3020 A1 = A : B1 = B : C1 = C
3030 M = SQR(A1) : N = B1/(2 * M) : P = C1 – N * N
3040 E = –N/M : IF E < –12 OR E > 12 THEN GOTO 3160
3045 IF 7 – P < 0 THEN GOTO 3160
3050 X4 = ((–1) * SQR (7 – P) – N)/M : X5 = (SQR (7 – P) – N)/M
3060 IF X4 > X5 THEN C = X4 : X4 = X5 : X5 = C
3070 IF X4 < –12 THEN X4 = –12
3080 IF X5 > 12 THEN X5 = 12
3090 FOR X = X4 TO X5 STEP 0.05
3100 Y = FNF(X)
3110 IF A < 0 THEN Y = –Y
3120 X3 = INT (X/0.1 + 0.5) : Y3 = INT (Y/0.1 + 0.5)
3130 IF 100 – Y3 <30 OR 100 – Y3 > 170 THEN GOTO 3150
3140 PSET (129 + X3, 100 – Y3)
3150 NEXT X
3160 RETURN
```

— En esta subrutina, primero se averigua si el coeficiente A es negativo, y en caso afirmativo se definen las variables A1, B1 y C1 en las que se almacenan los valores de A, B y C, respectivamente, pero cambiados de signo (línea **3010**). Hacer esto equivale «a dar la vuelta» a la parábola de coeficientes A, B y C.

— Si el valor de A es positivo, en las variables A1, B1 y C1 se almacenan los mismos valores de A, B y C, respectivamente (línea **3020**).

En la línea **3030** se calculan los valores de M, N y P en función de los valores de A1, B1 y C1. De este modo, la función $\mathbf{y = ax^2 + bx + c}$ se pone en la forma $\mathbf{y = (mx + n)^2 + p}$, y así se pueden aprovechar la mayoría de las instrucciones de la subrutina **3000** del programa anterior (representación gráfica de la función cuadrática $\mathbf{f(x) = (mx + n)^2 + p}$.

— El bloque de instrucciones **3040-3130** dibujan la parábola.

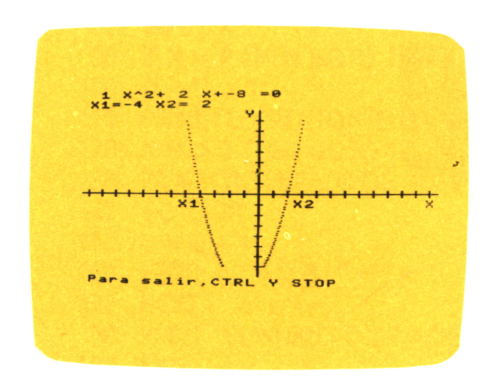
— La instrucción **3110** cambia de signo el valor de Y en el caso de que el coeficiente A de x^2 sea negativo. Esto se hace para que en la pantalla aparezca la verdadera parábola, ya que las asignaciones hechas en la línea **3010** producen el efecto de "dar la vuelta" a la misma.
La línea **3130** hace que no se representen en la pantalla los puntos de la parábola cuyas ordenadas sean mayores que 7 o menores que −7, respecto al orden de coordenadas elegido: (129,100).

Ejecuciones del programa

- Para hallar las soluciones de la ecuación $x^2 + 2x - 8 = 0$
  la instrucción **100** hay que escribirla así

  100 DATA 1, 2, –8

  Al ejecutar el programa se obtiene este resultado:

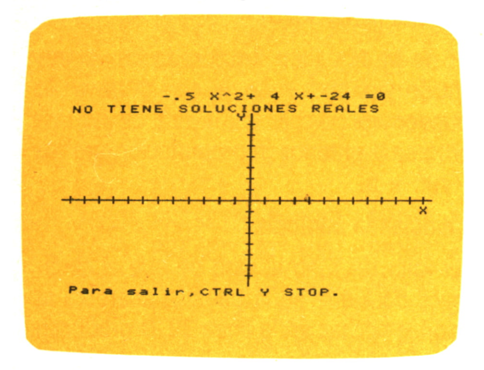


- Para hallar las soluciones de la ecuación $-\frac{1}{2}x^2 + 4x - 24 = 0$

  la instrucción **100** hay que escribirla así

  100 DATA –0.5, 4, –24

Al ejecutar el programa se obtiene este resultado:



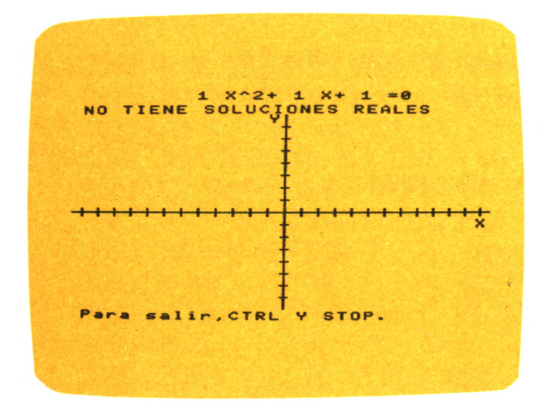
- Para hallar las soluciones de la ecuación

$$x^2 + x + 1 = 0$$

la instrucción **100** hay que escribirla así:

100 DATA 1, 1, 1

Al ejecutar el programa se obtiene este resultado:



5. Matemática básica

1. Las tablas de multiplicar

Presentación del problema

Se trata de practicar las tablas de multiplicar.

El programa

Se generan dos números al azar de una cifra y se pregunta cuál es su producto. Después de comparar la contestación con el resultado correcto, se anotan los aciertos y, en caso de error, se dan dos nuevas ocasiones. Se proponen 10 números utilizando un bucle (I = 1 a 10).

```
  1 REM MULTIPLICAR
 10 Z = RND (–TIME)
 20 C = 0 : D = 0
 30 FOR I = 1 TO 10
 40 A = INT(RND(1) * 9 + 1)
 50 B = INT(RND(1) * 9 + 1)
 60 P = A * B
 70 PRINT A; " * "; B; "="
 80 INPUT X
 90 IF X = P THEN GOTO 150
100 D = D + 1
110 PRINT "LO SIENTO: TE EQUIVOCASTE"
120 IF D > 2 THEN GOTO 160
130 PRINT "INTENTALO DE NUEVO"
140 GOTO 70
150 C = C + 1
160 PRINT A; "*"; B; "="; P
170 NEXT I
180 PRINT "EL NUMERO DE ACIERTOS ES"; C
190 END
```

La instrucción **10** hace que los números que generan las instrucciones **40** y **50** sean aleatorios. Eliminar la instrucción **10** si el microordenador no es MSX; en todo caso, sustituirla por la que corresponda al microordenador que se utilice. Con la instrucción **20** se inicializan los dos contadores C y D.

En las instrucciones **40, 50** y **60** se generan los números de una cifra A y B al azar, multiplicándolos a continuación y almacenando el resultado en P.

Se pide una contestación mediante la instrucción **80** y se compara en **90** con la solución correcta. Si la contestación es acertada, se suma 1 al contador C, en la **150,** y se escribe el resultado correcto. Si es errónea, se suma 1 al contador D, en la **100;** según el valor de D, se dan nuevas oportunidades (desde D = 0 hasta D = 2). El proceso se repite 10 veces.

Finalmente, en **180** se imprime el contenido final del contador C, el cual da el número de aciertos.

2. Resolución algebraica de la ecuación de segundo grado

Presentación del programa

Una ecuación de segundo grado $Ax^2 + Bx + C = 0$ tiene siempre dos soluciones, x_1 y x_2, dentro del cuerpo C de los números complejos. En el programa siguiente se van a obtener estas dos soluciones, indicando si son reales o complejas. Primeramente se introducen como datos los coeficientes A, B y C.

El programa

```
 1 REM SOLUCION DE UNA ECUACION DE SEGUNDO GRADO
10 INPUT A, B, C
20 D = B ↑ 2-4 * A * C
30 IF D < 0 THEN GOTO 90
```

En la instrucción **10** se leen los coeficientes A, B y C de la ecuación. La **20** asigna a D el valor del discriminante $B^2 - 4AC$. Por último, la instrucción **30** pregunta si el discriminante es negativo y, en tal caso, envía el control a la instrucción **90,** para calcular las soluciones complejas. Si no se verifica que $D < 0$, las raíces son reales y su cálculo comienza en la instrucción **40.**

```
40 X1 = (–B + SQR(D))/(2 * A)
50 X2 = (–B – SQR(D))/(2 * A)
60 PRINT "LAS SOLUCIONES SON REALES Y SUS"
70 PRINT "VALORES SON:"; X1; "Y"; X2
80 END
```

En las instrucciones **40** y **50** se calculan las dos soluciones reales X1 y X2 (que son iguales cuando D = 0). En la **70** se escriben las soluciones y en la **80** se para el programa.

```
 90  PR = –B/(2 * A)
100  PY = SQR(–D)/(2 * A) : PY = ABS(PY)
110  PRINT "LAS SOLUCIONES SON COMPLEJAS Y"
120  PRINT "SUS VALORES SON:"; PR; "+"; PI; "I"; "Y"; PR; "–";
      PY; "I"
```

La instrucción **90** calcula la parte real PR y la **100** la parte imaginaria PY. Como los dos números complejos son conjugados, basta con cambiar el signo de la parte imaginaria para obtener las soluciones en la instrucción **120.**

Ejecuciones del programa

- Si se ejecuta el programa, para A = 1, B = –4 y C = 3, se obtiene este resultado en la pantalla:

```
LAS SOLUCIONES SON REALES Y SUS
VALORES SON: 3 Y 1
```

- Si se ejecuta el programa, para A = 1, B = –2, C = 10, se obtienen dos soluciones complejas conjugadas:

```
LAS SOLUCIONES SON COMPLEJAS Y
SUS VALORES SON: 1 + 3I Y 1 – 3I.
```

3. Producto de polinomios

Presentación del programa

Para multiplicar dos polinomios P(x) y Q(x) los ordenamos previamente de mayor a menor grado.

$$P(x) = a_{p+1}x^p + a_px^{p-1} + \dots + a_2x + a_1$$
$$Q(x) = b_{q+1}x^1 + b_1x^{q-1} + \dots + b_2x + b_1$$

A continuación multiplicamos todos los monomios de P(x) por cada monomio de Q(x), tal como se muestra en este ejemplo:

$$P(x) = 2x^3 + x^2 + 3x - 1 \text{ y}$$
$$Q(x) = x^2 - 4x + 1$$

		2	1	3	–1
			1	–4	1
		2	1	3	–1
	–8	–4	–12	4	
2	1	3	–1		
2	–7	1	–12	7	–1

Luego el polinomio producto es:

$$R(x) = 2x^5 - 7x^4 + x^3 - 12x^2 + 7x - 1$$

Para escribir un programa que realice este producto se necesitan las siguientes variables:

P: grado del polinomio P(x)

A(P + 1): lista para guardar los P + 1 coeficientes de P(x)

Q: grado del polinomio Q(x)

B(Q + 1): lista para guardar los Q + 1 coeficientes de Q(x)

C(Q + 1, P + Q + 1): tabla para guardar las Q + 1 filas que se forman en los pasos intermedios de la multiplicación. Esta tabla tiene P + Q + 1 columnas pues este es el número de coeficientes del polinomio, producto R(x).

R(P + Q + 1): lista para guardar los P + Q + 1 coeficientes del polinomio producto R(x).

En el ejemplo anterior, tenemos:

P = 3 A (2, 1, 3, –1)
Q = 2 B (1, –4, 1)

```
           2   1   3  -1
               1  -4   1
        --------------------
           2   1   3  -1   }
      -8  -4 -12   4       } Q + 1 filas (2 + 1 = 3)
   2   1   3  -1           }
   -------------------------
   2  -7   1 -12   7  -1
   \_____________________/
P + Q + 1 columnas (3 + 2 + 1 = 6)
```

El programa

```
10  INPUT "GRADO DEL POLINOMIO P"; P
20  DIM A(P + 1)
30  FOR I = P + 1 TO 1 STEP - 1
40  PRINT "COEFICIENTE DEL TERMINO DE GRADO"; I - 1
50  INPUT A(I)
60  NEXT I
70  INPUT "GRADO DEL POLINOMIO Q"; Q
80  DIM B(Q + 1)
90  FOR I = Q + 1 TO 1 STEP - 1
100 PRINT "COEFICIENTE DEL TERMINO DE GRADO"; I - 1
110 INPUT B(I)
120 NEXT I
130 CLS
```

En el bloque de instrucciones **10-60** se introduce el grado del polinomio P(x) y sus coeficientes, de mayor grado a menor.

En el bloque de instrucciones **70-120** se realiza lo mismo que en el bloque anterior, pero en este caso para el polinomio Q(x).

```
140 DIM C (Q + 1, P + Q + 1)
150 FOR I = 1 TO Q + 1
160 FOR J = 1 TO P + 1
170 C(I, I + J - 1) = B(I) * A(J)
180 NEXT J
190 NEXT I
```

En este bloque se calculan los pasos intermedios de la multiplicación (la tabla C).

```
200  DIM R(P + Q + 1)
210  FOR K = 1 TO P + Q + 1
220  FOR I = 1 TO Q + 1
230  R(K) = R(K) + C(I, K)
240  NEXT I
250  NEXT K
```

Con estas instrucciones se calculan los coeficientes del polinomio producto R(x).

```
260  FOR K = P + Q + 1 TO 1 STEP - 1
270  PRINT "COEFICIENTE DEL TERMINO DE GRADO"; K - 1
280  PRINT R(K)
290  NEXT K
300  END
```

Este conjunto de instrucciones imprimen los coeficientes del polinomio producto.

Ejecución del programa

Si se ejecuta el programa para los polinomios: $P(x) = 2x^3 + x^2 + 3x - 1$

y $Q(x) = x^2 - 4x + 1$

se obtiene el siguiente resultado en la pantalla:

```
run
GRADO DEL POLINOMIO P? 3
COEFICIENTE DEL TERMINO DE GRADO 3
?2
COEFICIENTE DEL TERMINO DE GRADO 2
?1
COEFICIENTE DEL TERMINO DE GRADO 1
?3
COEFICIENTE DEL TERMINO DE GRADO 0
?-1
GRADO DEL POLINOMIO Q?2
COEFICIENTE DEL TERMINO DE GRADO 2
?1
COEFICIENTE DEL TERMINO DE GRADO 1
?-4
COEFICIENTE DEL TERMINO DE GRADO 0
?1
```

Al seguir la ejecución se borra la pantalla anterior y se obtiene el resultado siguiente:

```
COEFICIENTE DEL TERMINO DE GRADO 5
?2
COEFICIENTE DEL TERMINO DE GRADO 4
?–7
COEFICIENTE DEL TERMINO DE GRADO 3
?1
COEFICIENTE DEL TERMINO DE GRADO 2
?–12
COEFICIENTE DEL TERMINO DE GRADO 1
?7
COEFICIENTE DEL TERMINO DE GRADO 0
?–1
OK
■
```

4. Formación de las variaciones con repetición

Presentación del problema

- Las variaciones con repetición de M elementos, tomados de N en N, son los distintos grupos que se pueden formar de N elementos repetidos o no, tales que dos cualesquiera de ellos difieren en la naturaleza de algún elemento o en el orden de colocación de los mismos.

  Por ejemplo, con los elementos del conjunto {A, B, C} las variaciones con repetición de orden 3 que se pueden formar son las siguientes (en este ejemplo M = 3 y N = 3).

  M = 3
  N = 3

AAA	BAA	CAA
AAB	BAB	CAB
AAC	BAC	CAC
ABA	BBA	CBA
ABB	BBB	CBB
ABC	BBC	CBC
ACA	BCA	CCA
ACB	BCB	CCB
ACC	BCC	CCC

- El número total de variaciones con repetición, en este caso es:

  $VR_{3,3} = 3^3 = 27$

  En general: $\mathbf{VR_{M,N} = M^N}$

- La formación de las variaciones con repetición es independiente de los elementos. Si en el ejemplo anterior los elementos hubieran sido los del conjunto {1, 2, 3}, bastaría sustituir A por 1, B por 2 y C por 3.

  Luego a la hora de hacer el programa lo importante es el valor de M y N, pues al primer elemento se le asigna el 1, al segundo elemento se le asigna el 2, ..., y al n-ésimo elemento se le asigna N.

- El proceso a seguir para formar las variaciones con repetición es el que describimos a continuación (tomamos como ejemplo M = 4 y N = 3).

  1.° Se calcula el número de variaciones con repetición $V = M^N$.

  $$V = 4^3 = 64$$

  2.° Se hace variar I de 1 a V, para indicar la variación correspondiente.

  Haremos x = I − 1.
  Si dividimos x entre $N1^{N-1} = 4^2$ y a su parte entera le sumamos 1, tenemos el 1.er elemento: B(3).

  Si el resto lo dividimos entre $M^{N-2} = 4$ y a su parte entera le sumamos 1, tenemos el 2.° elemento: B(2).

  Si al resto le sumamos 1, tenemos el 3.er elemento: B(1).

I	x	B(3)	B(2)	B(1)
1	0	1	1	1
2	1	1	1	2
3	2	1	1	3
4	3	1	1	4
5	4	1	2	1
6	4	1	3	1
....	....	....	....	....

Como vemos, se van formando las variaciones con repetición.

3.° Para imprimir se realiza esta asignación:

B(3) 1.er elemento
B(2) 2.° elemento
B(1) 3.er elemento

El programa

```
10  INPUT "INTRODUCE LOS CARACTERES QUE DESEES VARIAR"; A$
20  M = LEN(A$)
30  INPUT "ORDEN DE LAS VARIACIONES"; N
40  DIM B(N)
50  V = M ↑ N
```

La línea **10** guarda en A$ los elementos con los que se desean formar las variaciones con repetición, y la **20,** el número M de dichos elementos.

La línea **30** pide el orden N de las variaciones.

La línea **40** dimensiona el vector B.

La línea **50** calcula el número de variaciones.

```
 60  FOR I = 1 TO V
 70  X = I - 1
 80  IF N = 1 THEN GOTO 140
 90  FOR J = N - 1 TO 1 STEP - 1
100  Z = X/(M ↑ J)
110  B(J + 1) = INT(Z) + 1
120  X = X-INT(Z) * (M ↑ J)
130  NEXT J
140  B(1) = X + 1
```

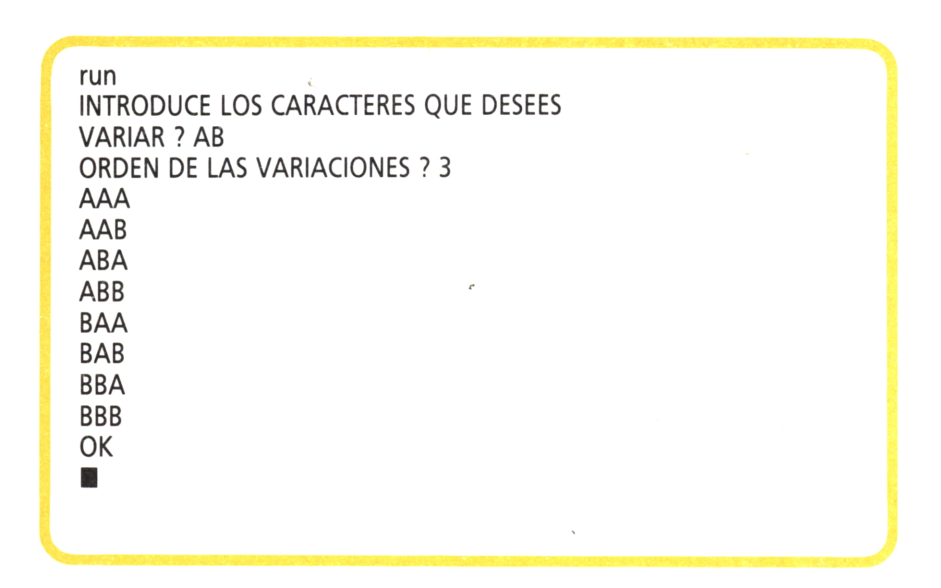
En este bloque de instrucciones se obtienen los valores de B(N), B(N–1), ..., B(1) que determinan los elementos de la variación I.

```
150  FOR K = N TO 1 STEP - 1
160  PRINT MID$(A$, B(K), 1);
170  NEXT K
180  PRINT
190  NEXT I
200  END
```

En este bucle se imprimen los elementos de la variación I.

Ejecuciones del programa

- Si se introducen como datos los elementos {A, B} y se hace N = 3 se obtiene este resultado:



```
run
INTRODUCE LOS CARACTERES QUE DESEES
VARIAR ? AB
ORDEN DE LAS VARIACIONES ? 3
AAA
AAB
ABA
ABB
BAA
BAB
BBA
BBB
OK
■
```

- Si se introducen como datos los elementos {R, S, T} y se hace N = 2 se obtiene este otro resultado:

```
run
INTRODUCE LOS CARACTERES QUE DESEES
VARIAR ? RST
ORDEN DE LAS VARIACIONES ?2
RR
RS
RT
SR
SS
ST
TR
TS
TT
OK
■
```

5. Formación de las variaciones ordinarias

Presentación del problema

Las variaciones ordinarias de M elementos, tomados de N en N, son los distintos grupos que se pueden formar de N elementos sin repetir, tales que dos cualesquiera de ellos difieran en la naturaleza de algún elemento o en el orden de colocación de los mismos.

Por ejemplo, con los elementos del conjunto {A, B, C, D} las variaciones ordinarias de orden 3 que se pueden formar son las siguientes (en este ejemplo M = 4 y N = 3).

ABC	BAC	CAB	DAB
ABD	BAD	CAD	DAC
ACA	BCA	CBA	DBA
ACD	BCD	CBD	DBC
ADB	BDA	CDA	DCA
ADC	BDC	CDB	DCB

El número total de variaciones ordinarias, en este caso, es:

$$V_{4,3} = 4.3.2 = 24$$

En general:

$$V_{M,N} = M.(M - 1) \ldots (M - N + 1)$$

El programa

El programa que forma las variaciones ordinarias se hace a partir del anterior, añadiendo algunas instrucciones para eliminar la repetición.

```
10  INPUT "INTRODUCE LOS CARACTERES QUE DESEES VARIAR"; A$
20  M = LEN(A$)
30  INPUT "ORDEN DE LAS VARIACIONES"; N
40  DIM B(N)
50  V = M ↑ N
```

La línea **10** guarda en A$ los elementos con los que se desea formar las variaciones ordinarias, y la **20** el número M de éstos.

La línea **30** pide el orden N de las variaciones.

La línea **40** dimensiona el vector B.

La línea **50** calcula el número de variaciones con repetición.

```
 60  FOR I = 1 TO V
 70  X = I - 1
 80  IF N = 1 THEN GOTO 140
 90  FOR J = N - 1 TO 1 STEP - 1
100  Z = X/(M ↑ J)
110  B(J + 1) = INT (Z) + 1
120  x = x - INT(Z) * (M ↑ J)
122  IF J = N - 1 THEN GOTO 130
124  FOR L = J + 2 TO N
126  IF B(J + 1) = B(L) THEN GOTO 190
128  NEXT L
130  NEXT J
140  B(1) = X + 1
```

En este bloque de instrucciones se obtienen los valores de B(N), B(N − 1), ..., B(1) que determinan los elementos de la variación I.

Las instrucciones **122, 124, 126** y **128** eliminan las repeticiones, saltando a otra variación (**GOTO 190** de la línea **126**).

```
142  IF N = 1 THEN GOTO 150
144  FOR L = 2 TO N
146  IF B(1) = B(L) THEN GOTO 190
148  NEXT L
```

Este bloque de instrucciones comprueba si B(1) es igual a alguno de los otros elementos del vector B, y en caso afirmativo elimina esta variación, saltando a la instrucción **190**.

```
150  FOR K = N TO 1 STEP - 1
160  PRINT MID$ (A$, B(K), 1);
170  NEXT K
180  PRINT
190  NEXT I
200  END
```

Este bucle imprime los elementos de la variación I.

Ejecución

- Si se introducen como datos los elementos del conjunto {A, B, C, D} y se hace N = 3 se obtiene este resultado.

```
run
INTRODUCE LOS CARACTERES QUE DESEES
VARIAR? ABCD
ORDEN DE LAS VARIACIONES ?3
ABC
ABD
ACB
ACD
ADB
ADC
BAC
...
...
...
DCB
OK
■
```

- Si el número de elementos coincide con el orden (M = N) se obtienen las permutaciones. Por ejemplo, para {1, 2, 3} y N = 3 se tiene:

```
run
INTRODUCE LOS CARACTERES QUE DESEES
VARIAR ? 123
ORDEN DE LAS VARIACIONES ?3
123
132
213
231
312
321
OK
■
```

6. Comercio

1. Tabla de amortización de un préstamo

Presentación del problema

Los pagos que hay que realizar anualmente a un Banco o a una entidad financiera para amortizar un préstamo es lo que se llama ***anualidades A de amortización.***

La fórmula de A es:

$$A = \frac{C \cdot I \cdot (1 + I)^T}{(1 + I)^T - 1}$$

donde:

C es el capital prestado por el Banco
I es el tanto por uno anual
T es el tiempo en años
A es la anualidad que hay que pagar para amortizar el préstamo.

El programa

El siguiente programa calcula el valor de A y forma una tabla con cuatro columnas que indican el número del pago (K), la parte del pago correspondiente al interés (X), la parte del pago correspondiente a la amortización (Y), y el capital amortizado hasta este pago (Z).

```
10  INPUT "CAPITAL"; C
20  INPUT "INTERES (EN % ANUAL)"; I
30  I = I/100
40  INPUT "TIEMPO (EN AÑOS)"; T
50  B = (1 + I) ↑ T
60  A = (C * I * B)/(B - 1)
70  A = INT (A + 0.5)
80  PRINT "LAS ANUALIDADES SON:"; A
```

Las líneas **10, 20** y **40** introducen los valores de C, I, T. La línea **30** transforma el tanto por ciento anual en tanto por uno.

Las líneas **50, 60** y **70** calculan el valor de A, redondeándolo sin decimales, y la línea **80** imprime su valor.

```
90  PRINT "PAGO"
100 PRINT TAB(4); "INTERESES"
110 PRINT TAB(13); "AMORTIZACION"
120 PRINT TAB(22); "CAPITAL AMORTIZADO"
```

Estas cuatro líneas imprimen las cabeceras de las cuatro columnas.

```
130 X = 0 : Y = 0 : Z = 0
140 FOR K = 1 TO T
150 X = INT ((C - Z) * I + 0.5)
160 Y = A - X
170 Z = Z + Y
180 PRINT K;
190 PRINT TAB(4); X;
200 PRINT TAB(13); Y;
210 PRINT TAB(22); Z
220 NEXT K
230 END
```

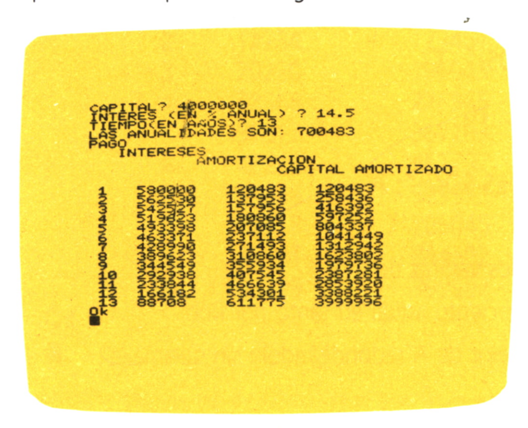
El bucle de variable K calcula e imprime los elementos de la tabla.

Ejecución del programa

Si se ejecuta el programa con los siguientes datos:

Capital prestado = 4 000 000
Interés anual = 14,5 %
Tiempo en años = 15

aparece en la pantalla el siguiente resultado:



2. Tabla de formación de un capital

Presentación del programa

Los pagos que se realizan, al principio de cada año, a un Banco o a una entidad financiera para formar un capital es lo que se llama anualidad A de capitalización.

La fórmula de A es:

$$A = \frac{C \,.\, I}{(1 + I)\,((1 + I)^T - 1)}$$

donde:

C es el capital que se quiere formar

I es el tanto por uno anual

T es el tiempo en años

A es la anualidad que hay que pagar, al principio de cada año, para formar el capital C.

El programa

El siguiente programa calcula el valor de A y forma una tabla de tres columnas que indican el número de pago (K), los intereses generados (X) por este pago, y los pagos anteriores y sus intereses, y el capital acumulado (Z) con este pago y los intereses.

```
10  INPUT "CAPITAL"; C
20  INPUT "INTERESES (EN % ANUAL)"; I
30  I = I/100
40  INPUT "TIEMPO (EN AÑOS)"; T
50  B = 1 + I
60  A = (C * I)/(B * (V ↑ T − 1))
70  A = INT(A + 0.5)
80  PRINT "LAS ANUALIDADES SON:"; A
```

Las líneas **10, 20** y **40** introducen los valores de C, I, T.

La línea **30** transforma el tanto por ciento anual en tanto por uno.

Las líneas **50, 60** y **70** calculan el valor de A redondeándolo sin decimales.

La línea **80** imprime el valor de A.

```
90  PRINT "PAGO"
100 PRINT TAB(4); "INTERESES"
110 PRINT TAB(13); "CAPITAL ACUMULADO"
120 X = 0 : Z = 0
130 FOR K = 1 TO T
140 X = INT ((Z + A) * I + 0.5)
150 Z = Z + A + X
160 PRINT K;
170 PRINT TAB(4); X;
180 PRINT TAB(13); Z
190 NEXT K
200 END
```

Las líneas **90, 100** y **110** imprimen las cabeceras de las tres columnas. El bucle de variable K calcula e imprime los elementos de la tabla.

Ejecución del programa

Si se desea formar un capital de un 1 000 000 de pesetas, en 10 años, y el interés que da el Banco es del 11,5 anual, las anualidades que hay que pagar al principio de cada año y la tabla de capitalización son las siguientes:

```
run
CAPITAL? 1 000 000
INTERES (EN % ANUAL)? 11.5
TIEMPO (EN AÑOS)? 10
LAS ANUALIDADES SON: 52356
         Pago  Intereses   Capital acumulado
           1      6021          58377
           2     12734         123467
           3     20220         196043
           4     28566         276965
           5     37872         367193
           6     48248         467797
           7     59818         579971
           8     72718         705045
           9     87101         844502
          10    103139         999997
OK
■
```

3. Elección de la variable en las anualidades de amortización y capitalización

Presentación del problema

En este programa se puede elegir entre anualidades de amortización (1) y anualidades de capitalización (2). Además se puede elegir la variable que se desea calcular:

(1) Capital C
(2) Interés I
(3) Tiempo T
(4) Anualidad A

Los cálculos se realizan por medio de las siguientes fórmulas y algoritmos:

1. *Anualidades de amortización*

1. Capital

$$C = \frac{A\,[(1 + I)^T - 1]}{I\,(1 + I)^T}$$

2. Interés

   1. Se estima el interés I = 10 %, por ejemplo, y se inicia la última estimación del interés I1 = 0.
   2. Se calcula una estimación de la anualidad

$$A1 = \frac{CI\,(1 + I)^T}{(1 + I)^T - 1}$$

   3. Se hace **X** = **ABS((I − I1)/2)**
      e **I1** = **I**
   4. Si A1 > A entonces I = I − X
      Si A1 < A entonces I = I + X
      (en ambos casos se vuelve a 2).
   5. Si A1 = A se imprime el interés I en % anual con dos decimales.

3. Tiempo

$$T = \frac{\text{LOG}\left(\dfrac{A}{A - CI}\right)}{\text{LOG}\,(1 + I)}$$

4. Anualidad

$$A = \frac{C \cdot I \cdot (1 + I)^T}{(1 + I)^T - 1}$$

2. *Anualidades de capitalización*

1. Capital

$$C = \frac{A(1 + I) [(1 + I)^T - 1]}{I}$$

2. Interés

    1. Se estima el interés I = 10 %, por ejemplo, y se inicia la última estimación del interés I1 = 0.
    2. Se calcula una estimación de la anualidad

$$A1 = \frac{C \cdot I}{(1 + I) [(1 + I)^T - 1]}$$

    3. Se hace **X** = ABS((I − I1)/2)
       e I1 = I
    4. Si A1 > A entonces I = I + X
       Si A1 < A entonces I = I − X
       (en ambos casos se vuelve a 2).
    5. Si A1 = A se imprime el interés I en % anual con dos decimales.

3. Tiempo

$$T = \frac{LOG \left[1 + \frac{C \cdot I}{A \cdot (1 + I)} \right]}{LOG (1 + I)}$$

4. Anualidad

$$A = \frac{C \cdot I}{(1 + I) [(1 + I)^T - 1]}$$

El programa

```
10 INPUT "AMORTIZACION (1), CAPITALIZACION (2)"; N
20 INPUT "VARIABLE: CAPITAL (1), INTERES (2), TIEMPO (3) y
   ANUALIDADES (4)"; M
30 IF M <> 1 THEN INPUT "CAPITAL"; C
40 IF M <> 2 THEN INPUT "INTERES (EN % ANUAL)"; I : I = I/100
50 IF M <> 3 THEN INPUT "TIEMPO (EN AÑOS)"; T
60 IF M <> 4 THEN INPUT "ANUALIDADES"; A
70 IF N = 1 THEN ON M GOSUB 1000, 2000, 3000, 4000
80 IF N = 2 THEN ON M GOSUB 5000, 6000, 7000, 8000
90 END
```

Ejecuciones del programa

- Si se desea calcular el interés I que corresponde a un préstamo de 2 000 000 de pesetas, con anualidades de 400 000 pesetas durante 10 años, basta seguir los pasos que muestra la pantalla:

```
run
AMORTIZACION(1), CAPITALIZACION(2) ?1
VARIABLE: CAPITAL(1), INTERES(2),
TIEMPO(3) Y ANUALIDADES(4) ?2
CAPITAL ? 2 000 000
TIEMPO (EN AÑOS) ? 10
ANUALIDADES? 400 000
INTERES (EN % ANUAL) = 15.1
OK
■
```

- Si se introducen los datos:
  (2) anualidades de capitalización,
  (3) tiempo,
  capital C = 3.000.000
  interés I = 16,5 %
  anualidad A = 900.000
  se obtiene el siguiente resultado en la pantalla:

```
run
AMORTIZACION(1), CAPITALIZACION(2) ?2
VARIABLE : CAPITAL(1), INTERES(2),
TIEMPO(3) Y ANUALIDADES(4) ?3
CAPITAL ? 3 000 000
INTERES (EN % ANUAL) ? 16.5
ANUALIDADES ? 900 000
TIEMPO = 2 AÑOS 6 MESES
OK
■
```

Nota: Realmente al cabo de dos años tendría 2.270.000 ptas. más las 900.000 pesetas de la anualidad correspondiente; es decir, 3.170.000 ptas. Luego el tiempo necesario es el que indica los años, despreciando los meses.

7. Estadística

1. Media, varianza y desviación típica

Presentación del problema

Si se tiene una serie de valores x_i y sus frecuencias absolutas n_i

x_i	n_i
x_1	n_1
x_2	n_2
.	.
.	.
.	.
x_k	n_k

La media aritmética, la varianza y la desviación típica vienen determinadas por las fórmulas siguientes:

$$m = \frac{\sum_{i=1}^{k} n_i x_i}{n}$$

$$V = \frac{\sum_{i=1}^{k} n_i (x_i - m)^2}{n}$$

$$s = \sqrt{V}$$

donde n es la suma de todas las frecuencias absolutas n_i.

El programa

```
10  INPUT "NUMERO DE VALORES DISTINTOS"; K
20  DIM X(K) : DIM N(K)
30  FOR I = 1 TO K
40  INPUT "VALOR, FRECUENCIA"; X(I), N(I)
50  NEXT I
```

En este grupo de instrucciones se leen los distintos valores y sus frecuencias absolutas.

```
 60  M = 0 : N = 0
 70  FOR I = 1 TO K
 80  M = M + N(I) * X(I)
 90  N = N + N(I)
100  NEXT I
110  M = M/N
120  PRINT "LA MEDIA ES:"
130  PRINT M
```

En estas instrucciones se calculan la suma de frecuencias N y la media aritmética M, y se imprime esta última.

```
140  V = 0
150  FOR I = 1 TO K
160  V = V + N(I) * (X(I) - M) ↑ 2
170  NEXT I
180  V = V/N
190  S = SQR(V)
```

De la línea **140** a la línea **180** se calcula la varianza V y en la línea **190** se calcula la desviación típica S.

```
200  PRINT "LA VARIANZA ES:"
210  PRINT V
220  PRINT "LA DESVIACION TIPICA ES:"
230  PRINT S
240  END
```

Estas instrucciones imprimen los valores de la varianza y la desviación típica.

Ejecución del programa

Si se ejecuta el programa para la siguiente distribución:

x_i	n_i
1	4
2	8
3	3
4	2
5	7
6	9

se obtienen estos resultados:

```
run
NUMERO DE VALORES DISTINTOS ? 6
VALOR, FRECUENCIA? 1,4
VALOR, FRECUENCIA? 2,8
VALOR, FRECUENCIA? 3,3
VALOR, FRECUENCIA? 4,2
VALOR, FRECUENCIA? 5,7
VALOR, FRECUENCIA? 6,9
LA MEDIA ES:
3.81818181
LA VARIANZA ES:
3.42148760
LA DESVIACION TIPICA ES:
1.84972636
OK
■
```

2. Distribución binomial

Presentación del problema

Si se realiza un experimento aleatorio n veces y los posibles resultados del experimento son dos, éxito y fracaso de probabilidades respectivas, cuyas probabilidades p y q = 1 − p, la distribución que se obtiene se llama ***binomial,*** de parámetros n y p:

$$B(n, p)$$

Por ejemplo, al lanzar una moneda al aire 20 veces (n = 20), los posibles resultados del experimento son dos, cara y cruz (éxito y fracaso), cuyas probabilidades son p = 0,5 y q = 1 − 0,5 = 0,5, respectivamente. Luego este experimento responde a una distribución binomial de parámetros 20 y 0,5:

$$B(20;0,5)$$

La ***variable binomial*** x de parámetros n y p es la variable que toma los valores 0, 1, 2, ... n (número de éxitos)

$$x = 0, 1, 2, ..., n$$

con probabilidad:

$$P_x = \frac{n!}{x!\,(n-x)!} \cdot P^x \cdot (1 - P)^{n-x}$$

La media es: $M = n \cdot p$

y la varianza es: $V = n \cdot p \cdot (1 - p)$

En el ejemplo anterior la probabilidad de obtener, por ejemplo, 5 caras es:

$$P_5 = \frac{20!}{5!\ 15!} \cdot 0,5^5\ 0,5^{15}$$

la media es: $M = 20 \cdot 0,5 = 10$

la varianza: $V = 20 \cdot 0,5 \cdot 0,5 = 5$

y la desviación típica: $S = \sqrt{5}$

El programa

```
10  INPUT "PARAMETROS : N, P"; N, P
20  INPUT "UN VALOR(1) O TODA LA DISTRIBUCION(2)"; A
30  PRINT "VALOR X", "PROBABILIDAD PX"
40  IF A = 1 THEN INPUT "VALOR"; X : GOSUB 1000
50  IF A = 2 THEN FOR X = 0 TO N : GOSUB 1000 : NEXT X
60  END
```

Este es el bloque principal del programa, en el que se introducen los valores de los parámetros N y P, y se elige entre calcular la probabilidad de un solo valor (A = 1) o toda la distribución (A = 2).

Se llama también a la subrutina **1000** para que calcule la probabilidad.

```
1000  Z = N : GOSUB 2000 : C = Y
1010  Z = X : GOSUB 2000 : D = Y
1020  Z = N - X : GOSUB 2000 : E = Y
1030  PX = C/(D * E) * P ↑ X * (1 - P) ↑ (N - X)
1040  PRINT X, PX
1050  RETURN
```

En esta subrutina se calcula la probabilidad PX del valor X. Se hacen tres llamadas a la subrutina **2000** para que calcule los factoriales de N, X y N − X.

```
2000  Y = 1
2010  FOR I = 1 TO Z
2020  Y = Y * I
2030  NEXT I
2040  RETURN
```

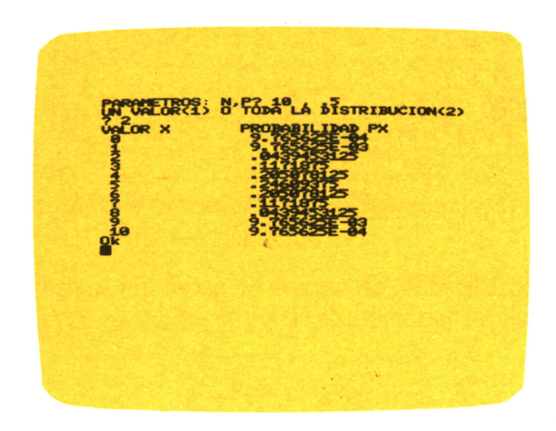
Esta subrutina calcula el factorial de Z, devolviendo como resultado Y.

Ejecuciones del programa

- Si se dan como datos N = 20, P = 0.5 y X = 5 se obtiene este resultado:

```
run
PARAMETROS : N, P? 20, 0.5
UN VALOR (1) O TODA LA DISTRIBUCION (2)
?1
VALOR X          PROBABILIDAD PX
VALOR?5
5                0.01478577
OK
■
```

- Si se desea obtener todas las probabilidades de la distribución B (10, 0.5) basta con seguir los pasos que muestra la pantalla:



3. Distribuciones bidimensionales

Presentación del problema

Una distribución bidimensional contempla el caso de una población de n elementos y en cada individuo de la misma considera dos características X e Y.

Ejemplo: Una población de 10 personas de las que se consideran la edad (X) y el peso (Y).

Sean: $X_1, X_2, \ldots, X_j, \ldots, X_p$ el conjunto de valores que puede tomar X, e $Y_1, Y_2, \ldots, Y_j, \ldots, Y_q$ el conjunto de valores que puede tomar Y.

La distribución de frecuencias absolutas de los n elementos de la población se expresa en una tabla como ésta:

Y \ X	X_1	X_2	...	X_i	...	x_p
Y_1	n_{11}	n_{21}	...	n_{i1}	...	n_{p1}
Y_2	n_{12}	n_{22}	...	n_{i2}	...	n_{p2}
⋮	⋮	⋮	...	⋮	...	⋮
Y_j	n_{1j}	n_{2j}	...	n_{ij}	...	n_{pj}
⋮	⋮	⋮	...	⋮	...	⋮
Y_q	n_{1q}	n_{2q}	...	n_{iq}	...	n_{pq}

$$n = \sum_{i=1}^{p} \sum_{j=1}^{q} n_{ij}$$

La tabla de la distribución de frecuencias relativas f_{ij} se obtiene dividiendo cada frecuencia absoluta n_{ij} entre n.

Y \ X	X_1	X_2	...	X_i	...	X_p	
Y_1	f_{11}	f_{21}	...	f_{i1}	...	f_{p1}	h_1
Y_2	f_{12}	f_{22}	...	f_{i2}	...	f_{p2}	h_2
. . .	. . .	. . .	...	. . .	...	. . .	. . .
Y_j	f_{1j}	f_{2j}	...	f_{ij}	...	f_{pj}	h_j
. . .	. . .	. . .	... ...	. . .	... ...	. . .	. . .
Y_q	f_{1q}	f_{2q}	...	f_{iq}	...	f_{pq}	h_q
	g_1	g_2	...	g_i	...	g_p	1

Cada g_i es la suma de los elementos de la columna i

$$g_i = \sum_{j=1}^{q} f_{ij}$$

y cada h_j es la suma de los elementos de la línea j

$$h_j = \sum_{i=1}^{p} f_{ij}$$

A una distribución bidimensional se le pueden asociar dos distribuciones que se llaman ***marginales.***

— ***Distribución marginal respecto de X,*** que asocia a cada X_i el número g_i.

— ***Distribución marginal respecto de Y,*** que asocia a cada Y_j el número h_j.

La media y la varianza de cada una de estas distribuciones marginales son:

$$MX = \sum_{i=1}^{p} g_i X_i \qquad VX = \sum_{i=1}^{p} g_i (x_i - MX)^2$$

$$MY = \sum_{j=1}^{q} h_j Y_j \qquad VY = \sum_{j=1}^{p} h_j (Y_j - MY)^2$$

Para medir la variación conjunta de las dos variables X e Y se define la covarianza de la distribución bidimensional:

$$CV = \sum_{i=1}^{p} \sum_{j=1}^{q} f_{ij}(X_i - M_X) \cdot (Y_j - MY)$$

La relación entre las variables X e Y viene dada por las ecuaciones de las ***rectas de regresión*** de Y sobre X y de X sobre Y cuyas gráficas se ajustan a las nubes formadas por los n puntos (X_i, Y_i) e (Y_i, X_i), respectivamente.

Recta de regresión de Y sobre X:

$$Y - MY = \frac{CV}{VX} (X - MX)$$

Coeficiente de regresión:

$$BY = \frac{CV}{VX}$$

Recta de regresión de X sobre Y:

$$X - MX = \frac{CV}{VY} (Y - MY)$$

Coeficiente de regresión:

$$BX = \frac{CV}{VY}$$

Una vez ajustada una recta de regresión a la nube de puntos se define el ***coeficiente de correlación*** R entre los variables X e Y.

$$R = \frac{CV}{\sqrt{VX} \cdot \sqrt{VY}}$$

que sirve para determinar la ***dependencia*** entre las dos variables.

El programa

```
10  INPUT "NUMERO DE VALORES DE X"; P
20  DIM X(P)
30  FOR I = 1 TO P
40  PRINT "VALOR DE X"; I
50  INPUT X(I)
60  NEXT I
70  CLS
```

En este bloque de instrucciones se introducen los distintos valores de X.

```
80  INPUT "NUMERO DE VALORES DE Y"; Q
90  DIM Y(Q)
100 FOR J = 1 TO Q
110 PRINT "VALOR DE Y"; J
120 INPUT Y(J)
130 NEXT J
140 CLS
```

En este bloque de instrucciones se introducen los distintos valores de Y.

```
150 DIM N(P, Q) : N = 0
160 FOR I = 1 TO P
170 FOR J = 1 TO Q
180 PRINT "FRECUENCIA DE X"; I; ",Y"; J
190 INPUT N(I, J)
200 N = N + N(I, J)
210 NEXT J
220 NEXT I
230 CLS
```

En este conjunto de instrucciones se introducen los valores de las frecuencias absolutas n_{ij} y se calcula su suma total n.

```
240 DIM F(P, Q)
250 FOR I = 1 TO P
260 FOR J = 1 TO Q
270 F(I, J) = N(I, J)/N
280 NEXT J
290 NEXT I
```

En estas instrucciones se calculan las frecuencias relativas f_{ij}.

```
300 DIM G(P) : DIM H(Q)
310 FOR I = 1 TO P
320 FOR J = 1 TO Q
330 G(I) = G(I) + F(I, J)
340 H(J) = H(J) + F(I, J)
350 NEXT J
360 NEXT I
```

En estas instrucciones se calculan las frecuencias g_i y h_i de las distribuciones marginales X e Y, respectivamente.

```
370  MX = 0 : MY = 0 : VX = 0 : VY = 0 : CV = 0
380  FOR I = 1 TO P
390  MX = MX + G(I) * X(I)
400  NEXT
```

Las líneas **380-400** calculan la media MX.

```
410  FOR J = 1 TO Q
420  MY = MY + H(J) * Y(J)
430  NEXT J
```

Las líneas **410-430** calculan la media MY.

```
440  FOR I = 1 TO P
450  VX = VX + G(I) * (X(I) − MX) ↑ 2
460  NEXT I
```

Las líneas **440-460** calculan el valor de la varianza VX.

```
470  FOR J = 1 TO Q
480  VY = VY + H(J) * (Y(J) − MY) ↑ 2
490  NEXT J
```

Las líneas **470-490** calculan el valor de la varianza VY.

```
500  FOR I = 1 TO P
510  FOR J = 1 TO Q
520  CV = CV + F(I, J) * (X(I) − MX) * (Y(J) − MY)
530  NEXT J
540  NEXT I
```

Las líneas **500-540** calculan el valor de la covarianza CV.

```
550  BY = CV/VX
560  BX = CV/VY
570  R = CV/(SQR(VX) * SQR(VY))
```

La línea **550** calcula el coeficiente de regresión BY de la recta de Y sobre X. La línea **560** calcula el coeficiente de regresión BX de la recta de X sobre Y. La línea **570** calcula el coeficiente de correlación R.

```
580 DEF FNR(Z) = INT(Z * 100 + 0.5)/100
590 MX = FNR(MX)
600 MY = FNR(MY)
610 VX = FNR(VX)
620 VY = FNR(VY)
630 CV = FNR(CV)
640 BY = FNR(BY)
650 BX = FNR(BX)
660 R = FNR(R)
```

La línea **580** define la función FNR(Z) que redondea a dos decimales el valor de Z.

Las líneas **590-660** utilizan esta función para redondear a dos decimales todos los valores obtenidos.

```
670 PRINT "VARIABLE MARGINAL X"
680 PRINT "MEDIA ="; MX
690 PRINT "VARIANZA ="; VX
700 PRINT "VARIABLE MARGINAL Y"
710 PRINT "MEDIA ="; MY
720 PRINT "VARIANZA ="; VY
730 PRINT "COVARIANZA"
740 PRINT CV
```

Estas instrucciones imprimen las ***medias*** y ***varianzas*** de las distribuciones marginales, y la ***covarianza.***

```
750 PRINT "RECTA DE REGRESION DE Y SOBRE X"
760 PRINT "Y"; –MY; "="; BY; "* (X"; –MX; ")"
770 PRINT "RECTA DE REGRESION DE X SOBRE Y"
780 PRINT "X"; –MX; "="; BX; "* (Y"; –MY; ")"
790 PRINT "COEFICIENTE DE CORRELACION"
800 PRINT R
810 END
```

Este conjunto de instrucciones imprimen las ***ecuaciones*** de ***las dos rectas de regresión*** y el valor del ***coeficiente de correlación.***

Ejecución del programa

Sea la tabla de distribución de frecuencias:

y \ x	1	2	3	4	5
2	2	1	3	5	2
4	4	2	1	3	4
6	1	6	4	0	6
8	3	4	2	7	5

Al ejecutar el programa y al ir introduciendo estos datos se obtienen sucesivamente las siguientes pantallas:

```
run
NUMERO DE VALORES DE X ? 5
VALOR DE X1
?1
VALOR DE X2
?2
VALOR DE X3
?3
VALOR DE X4
?4
VALOR DE X5
?5
```

```
NUMERO DE VALORES DE Y?4
VALOR DE Y1
?2
VALOR DE Y2
?4
VALOR DE Y3
?6
VALOR DE Y4
?8
```

```
FRECUENCIA DE X1, Y1
?2
FRECUENCIA DE X1, Y2
?4
FRECUENCIA DE X1, Y3
?1
FRECUENCIA DE X1, Y4
?3
FRECUENCIA DE X2, Y1
?1
FRECUENCIA DE X2, Y2
?2
FRECUENCIA DE X2, Y3
?6
FRECUENCIA DE X2, Y4
?4
FRECUENCIA DE X3, Y1
?3
FRECUENCIA DE X3, Y2
?1
FRECUENCIA DE X3, Y3
?4
FRECUENCIA DE X3, Y4
?2
FRECUENCIA DE X4, Y1
?5
FRECUENCIA DE X4, Y2
?3
FRECUENCIA DE X4, Y3
?0
FRECUENCIA DE X4, Y4
?7
FRECUENCIA DE X5, Y1
?2
FRECUENCIA DE X5, Y2
?4
FRECUENCIA DE X5, Y3
?6
FRECUENCIA DE X5, Y4
?5
```

```
VARIABLE MARGINAL X
MEDIA = 3.25
VARIANZA = 2.03
VARIABLE MARGINAL Y
MEDIA = 5.42
VARIANZA = 5.01
COVARIANZA
0.08
RECTA DE REGRESION DE Y SOBRE X
Y - 5.42 = 0.04 * (X - 3.25)
RECTA DE REGRESION DE X SOBRE Y
X - 3.25 = 0.02 * (Y - 5.42)
COEFICIENTE DE CORRELACION
0.03
OK
■
```

Al ser el coeficiente de correlación casi cero (R = 0,03) las dos variables apenas si están relacionadas.

Como los coeficientes de regresión son prácticamente cero, el ajuste es malo.

Observación

Dado que en el programa la tabla N(I, J) sólo se utiliza para calcular la tabla F(I, J), se puede ahorrar bastante memoria sustituyendo ambas por la segunda. Para ello basta con modificar las líneas siguientes del programa:

```
150  DIM F(P, Q) : N = 0
190  INPUT F(I, J)
200  N = N + F(I, J)
240  (se suprime esta línea)
270  F(I, J) = F(I, J)/N
```

De esta forma, en las líneas **150-230**, F(I, J) representa las frecuencias absolutas.

En la línea **270** se realiza el cambio a frecuencias relativas, y a partir de esta línea F(I, J) representa las frecuencias relativas.

8. Matemática avanzada

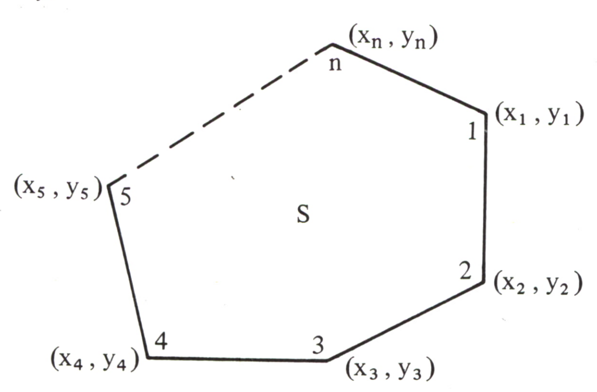
1. Área de un polígono

Presentación del problema

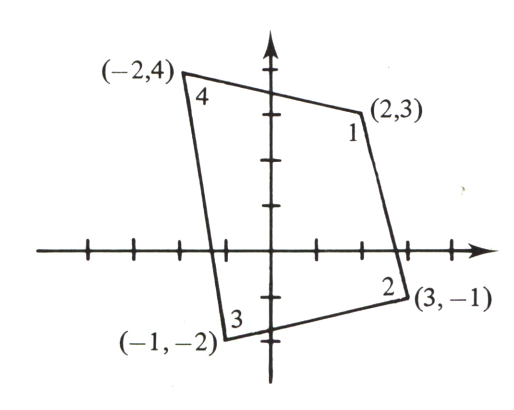
El área de un polígono, en función de sus vértices (x_i, y_i), viene dada por la fórmula:

$$S = \frac{\left| \sum_{i=2}^{n} (x_1 - x_i) \cdot (y_{i+1} - y_{i-i}) \right|}{2}$$

donde $y_{n+1} = y_1$



Por ejemplo, el área de este cuadrilátero



se obtiene de la siguiente forma:

$$S = \frac{|(x_1 - x_2) \cdot (y_3 - y_1) + (x_1 - x_3) \cdot (y_4 - y_2) + (x_1 - x_4) \cdot (y_1 - y_3)|}{2} =$$

$$= \frac{|(2 - 3) \cdot (-1 - 3) + (2 + 1) \cdot (4 + 1) + (2 + 2) \cdot (3 + 2)|}{2} =$$

$$= \frac{|5 + 15 + 20|}{2} = \frac{40}{2} = 20$$

El programa

```
10  INPUT "NUMERO DE VERTICES"; N
20  DIM X(N) : DIM Y(N + 1)
30  FOR I = 1 TO N
40  PRINT T "COORDENADAS DEL VERTICE"; I
50  INPUT X(I), Y(I)
60  NEXT I
70  Y(N + 1) = Y(1)
```

En este grupo de instrucciones se introduce el número de vértices y sus valores.

```
 80  S = 0
 90  FOR I = 2 TO N
100  S = S + (X(1) - X(I)) * (Y(I + 1) - Y(Y - 1))
110  NEXT I
120  S = ABS(S)/2
130  PRINT "EL AREA DEL POLIGONO ES:"
140  PRINT S
150  END
```

El bloque de instrucciones **80-120** calcula el área, y las instrucciones **130** y **140** imprimen el resultado.

Ejecución del programa

Si se ejecuta el programa para los datos N = 4, (2,3), (3,–1), (–1,–2) y (–2,4) se obtiene el siguiente resultado en la pantalla:

```
run
NUMERO DE VERTICES? 4
COORDENADAS DEL VERTICE 1
? 2, 3
COORDENADAS DEL VERTICE 2
? 3, -1
COORDENADAS DEL VERTICE 3
? -1, -2
COORDENADAS DEL VERTICE 4
? -2, 4
EL AREA DEL POLIGONO ES:
20
OK
■
```

2. Operaciones con vectores

Presentación del problema

Vamos a estudiar un programa que permita, conociendo las componentes de dos vectores, resolver las siguientes operaciones entre dichos vectores:

a) suma

b) resta

c) producto escalar

d) producto vectorial

teniendo en cuenta que si las componentes del vector $\vec{A}$ son (A1, A2, A3) y las del vector $\vec{B}$ (B1, B2, B3), los resultados de estas operaciones se obtienen a partir de las expresiones:

$\vec{A} + \vec{B}$ = (A1 + B1, A2 + B2, A3 + B3)
$\vec{A} - \vec{B}$ = (A1 − B1, A2 − B2, A3 − B3)
$\vec{A} \cdot \vec{B}$ = A1 * B1 + A2 * B2 + A3 * B3 (producto escalar)
$\vec{A} \times \vec{B}$ = (A2 * B3 − A3 * B2, A3 * B1 − A1 * B3, A1 * B2 − A2 * B1) (producto vectorial)

El programa

```
1  REM OPERACIONES CON VECTORES
10 PRINT "COORDENADAS VECTOR A"
20 INPUT A1, A2, A3
30 PRINT "COORDENADAS DEL VECTOR B"
40 INPUT B1, B2, B3
50 PRINT
60 PRINT "A + B = ("; A1 + B1; "."; A2 + B2; "'"; A3 + B3;")"
70 PRINT "A - B = ("; A1 - B1; ","; A2 - B2; ","; A3 - B3;")"
80 PRINT "A · B ="; A1 * B1 + A2 * B2 + A3 * B3
90 PRINT "A*B=(";A2*B3-A3*B2;",";A3*B1*-A1*B3;
   ",";A1*B2-A2*B1;")"
```

En estas instrucciones, después de introducir las coordenadas de los vectores (instrucciones **10** a **40**), se obtienen e imprimen los resultados de las distintas operaciones (instrucciones **60** a **90**).

```
100 PRINT "MAS DATOS (1 = SI, 0 = NO)"
110 INPUT X
120 IF X = 1 THEN GOTO 10
130 END
```

En esta última parte se pregunta si se introducirán nuevos datos y, en caso afirmativo, se inicia el proceso de nuevo. En caso contrario, se termina el programa.

Ejecución del programa

Si se introducen como datos los vectores $\vec{A}(2, 3, 1)$ y $\vec{B}(-1, 0, 2)$ se obtiene como resultado lo que muestra la pantalla:

```
A + B = (1, 3, 3)
A - B = (3, 3, -1)
A · B = 0
A * B = (6, -5, 3)
```

3. Módulos, cosenos directores y ángulo de dos vectores

Presentación del problema

Dados dos vectores $\vec{A}$ y $\vec{B}$, si (A1, A2, A3) son las coordenadas del primer vector $\vec{A}$ y (B1, B2, B3), las coordenadas del segundo vector $\vec{B}$, son módulos respectivos M1 y M2, se obtienen aplicando estas fórmulas:

$$M1 = \sqrt{A1^2 + A2^2 + A3^2}$$

$$M2 = \sqrt{B1^2 + B2^2 + B3^2}$$

Los cosenos del ángulo que forman con los ejes se obtienen calculando estos cocientes:

$$\frac{A1}{M1} \qquad \frac{A2}{M1} \qquad \frac{A3}{M1}$$

$$\frac{B1}{M2} \qquad \frac{B2}{M2} \qquad \frac{B3}{M2}$$

Y el coseno del ángulo que forman se determina así:

$$\cos(\widehat{\vec{A}, \vec{B}}) = \frac{A1 * B1 + A2 * B2 + A3 * B3}{M1 * M2}$$

Los vectores son perpendiculares si este coseno es cero, y son paralelos si su valor es 1 ó –1.

Observación. Para calcular el ángulo $Z = (\widehat{\vec{A}, \vec{B}})$ se utiliza la fórmula:

$$Z = \text{arc tg } \frac{\sqrt{1 - C^2}}{C} \quad \text{siendo } C = \cos(\widehat{\vec{A}, \vec{B}})$$

Esto es necesario tenerlo en cuenta porque algunos microordenadores no disponen de la función arco coseno.

El programa

a) El módulo de cada uno de ellos.
b) Los cosenos de los ángulos que forman con los ejes X, Y, Z (cosenos directores).
c) El ángulo que forman los dos vectores.

El programa averigua, además, si los vectores son paralelos o perpendiculares.

```
10  PRINT "COORDENADAS DEL VECTOR A"
20  INPUT A1, A2, A3
30  PRINT "COORDENADAS DEL VECTOR B"
40  INPUT B1, B2, B3
```

En estas líneas se han introducido las coordenadas de los vectores $\vec{A}$ y $\vec{B}$.

```
50  M1 = SQR(A1 ↑ 2 + A2 ↑ 2 + A3 ↑ 2)
60  M2 = SQR(B1 ↑ 2 + B2 ↑ 2 + B3 ↑ 2)
70  PRINT "MODULOS" : PRINT M1 : PRINT M2
```

En las líneas **50** y **60** se calculan los módulos de los dos vectores, y en la línea **70** se imprimen.

```
 80  IF M1 = 0 THEN GOTO 110
 90  PRINT "COSENOS DIRECTORES DEL VECTOR A"
100  PRINT A1/M1 : PRINT A2/M1 : PRINT A3/M1
110  IF M2 = 0 THEN GOTO 140
120  PRINT "COSENOS DIRECTORES DEL VECTOR B»
130  PRINT B1/M2 : PRINT B2/M2 : PRINT B3/M2
```

En el bloque de instrucciones **80-130** se calculan los cosenos directores de los dos vectores, $\vec{A}$ y $\vec{B}$.

```
140  IF M1 = 0 OR M2 = 0 THEN PRINT "NO HAY ANGULO" : GOTO
     240
150  C = (A1 * B1 + A2 * B2 + A3 * B3) / (M1 * M2)
160  C = INT (C * 1000 + 0.5)/1000
170  IF C = 0 THEN Z = 90 : GOTO 200
180  Z = ATN (SQR (1 - C ↑ 2) / C)
190  Z = Z * 180/3.1416
200  PRINT "ANGULO ENTRE LOS DOS VECTORES"
210  PRINT Z; "GRADOS"
220  IF C = 0 THEN PRINT "LOS VECTORES SON PERPENDICULARES"
230  IF C = 1 OR C = -1 THEN PRINT "LOS VECTORES SON PARA-
     LELOS"
```

La línea **150** calcula el coseno C del ángulo que forman y la línea **160** lo redondea a 3 decimales.

La línea **170** comprueba si el valor de C es cero, para evitar en la línea **180** se produzca un error al calcular el ángulo Z por medio de la función arcotangente.

La línea **190** pasa el ángulo Z de radianes a grados sexagesimales.

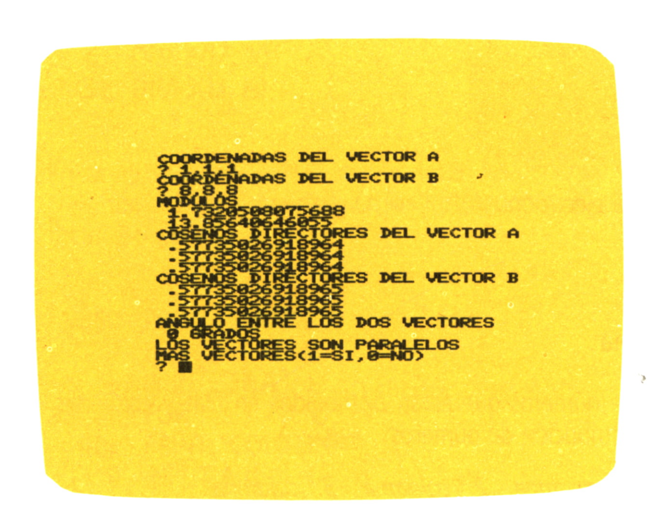
Las líneas **200-230** imprimen el valor del ángulo e informes en el caso de que los vectores sean perpendiculares o paralelos.

```
240 PRINT "MAS VECTORES (1 = SI, 0 = NO)"
250 INPUT I
260 IF I = 1 THEN GOTO 10
270 END
```

En esta última parte se pregunta si hay más vectores, para reiniciar o no el proceso de cálculo.

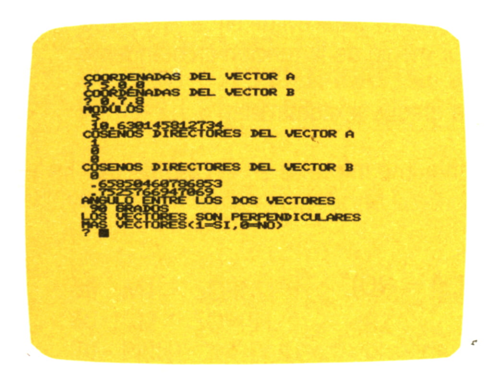
Ejecuciones del programa

- Si se introducen como datos los vectores $\vec{A}(1, 1, 1)$ y $\vec{B}(8, 8, 8)$, paralelos, se obtiene el siguiente resultado:



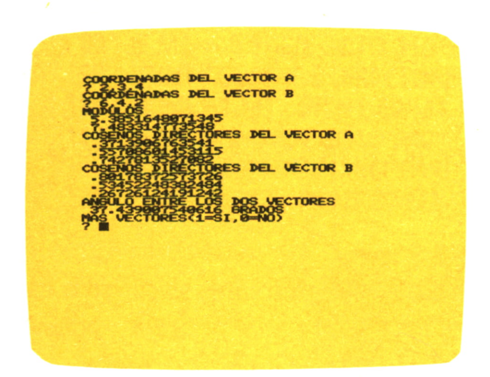
Al pulsar 0 termina la ejecución.

- Si se introducen como datos los vectores $\vec{A}(5, 0, 0)$ y $\vec{B}(0, 7, 8)$ perpendicular, se obtiene el siguiente resultado:



Al introducir 0 termina la ejecución.

- Si se introducen como datos los vectores $\vec{A}(2, 3, 4)$ y $\vec{B}(6, 4, 2)$ se obtiene el siguiente resultado:



Al introducir 0 termina la ejecución.

4. Suma de matrices

Presentación del problema

Una matriz es una tabla de números que están distribuidos en filas y columnas. Por ejemplo, la siguiente distribución de números:

	1.ª columna	*2.ª columna*	*3.ª columna*
1.ª fila	2	3	4
2.ª fila	1	0	–2

es una matriz de 2 filas y 3 columnas. Si a la matriz anterior se le llama A, el elemento que ocupa la primera fila y la segunda columna, que es 3, suele indicarse a_{12}, y en trabajos de programación, A(1, 2). El primer número indica la fila y el segundo la columna. Luego los elementos de la matriz A son:

A (1, 1) = 2 A (1, 2) = 3 A (1, 3) = 4
A (2, 1) = 1 A (2, 2) = 0 A (2, 3) = –2

La matriz A se dice que es de dimensión 2 × 3, pues tiene 2 filas y 3 columnas. Para sumar dos matrices A y B, ambas tienen que ser de la misma dimensión (igual número de filas e igual número de columnas). La matriz suma se obtiene sumando los términos que ocupan la misma posición. Ejemplo:

$$\begin{pmatrix} 2 & 3 & 4 \\ 1 & 0 & -2 \end{pmatrix} + \begin{pmatrix} 6 & 3 & 1 \\ -4 & 2 & 5 \end{pmatrix} = \begin{pmatrix} 8 & 6 & 5 \\ -3 & 2 & 3 \end{pmatrix}$$

El programa

En el programa, primeramente se debe indicar la dimensión de las mismas, M y N. A continuación, hay que definir las matrices A(M, N); B(M, N); S(M, N); siendo A-B las matrices sumandos (datos) y S la matriz suma.

```
 1  REM SUMA DE MATRICES
10  INPUT "FILAS", M
20  INPUT "COLUMNAS", N
30  DIM A(M, N)
40  DIM B(M, N)
50  DIM S(M, N)
```

En las instrucciones **INPUT** se introducen las dimensiones (M = número de filas y N = números de columnas). En las instrucciones **30, 40** y **50** se dimensionan las matrices A, B y S.

```
 60  PRINT "MATRIZ A"
 70  FOR I = 1 TO M
 80  PRINT "FILA"; I
 30  FOR J = 1 TO N
100  PRINT "COLUMNA"; J
110  INPUT A (I, J)
120  NEXT J
130  NEXT I
140  CLS
```

Mediante el bucle **70-130** se hace variar la fila y mediante el **90-120,** la columna. La instrucción **110** va pidiendo los distintos valores de la matriz A. La **140** borra la pantalla.

```
150 PRINT "MATRIZ B"
160 FOR I = 1 TO M
170 PRINT "FILA"; I
180 FOR J = 1 TO N
190 PRINT "COLUMNA"; J
200 INPUT B (I, J)
210 NEXT J
220 NEXT I
230 CLS
```

Este bloque repite el mismo proceso que el anterior, sólo que ahora se introducen los valores de la matriz B.

```
240 FOR I = 1 TO M
250 FOR J = 1 TO N
260 S (I, J) = A (I, J) + B (I, J)
270 NEXT J
280 NEXT I
```

Por medio de los bucles **240-280** (que hacen variar las filas) y **250-270** (que hacen variar las columnas) se suman los elementos que ocupan la misma posición en A y B (instrucción **260**), almacenándose el resultado en S(I, J).

```
290 PRINT "LA MATRIZ SUMA ES"
300 FOR I = 1 TO M
310 PRINT "FILA"; I
320 FOR J = 1 TO N
330 PRINT S(I, J)
340 NEXT J
350 NEXT I
360 END
```

En este grupo de instrucciones se imprime la matriz suma S por filas, dándose por terminado el programa.

Ejecución del programa

Si se ejecuta el programa para M = 2 y N = 3, siendo las matrices

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 1 & 0 & -2 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 3 & 1 \\ -4 & 2 & 5 \end{pmatrix}$$

se obtiene este resultado:

```
FILA 1
  8
  6
  5
FILA 2
 -3
  2
  3
```

En consecuencia, la matriz suma es:

$$S = \begin{pmatrix} 8 & 6 & 5 \\ -3 & 2 & 3 \end{pmatrix}$$

5. Producto de matrices

Presentación del problema

Para poder multiplicar dos matrices A y B es necesario que el número de columnas de A coincida con el número de filas de B.

Si la dimensión de A es M × N, la dimensión de B tiene que ser N × P; es decir, el número de columnas de la primera matriz tiene que coincidir con el número de filas de la segunda.

La matriz producto R = A × B tiene por elementos:

R(1, 1) = suma de los productos de los términos de la 1.ª fila de A por los de la 1.ª columna de B.

R(1, 2) = suma de los productos de los términos de la 1.ª fila de A por los de la 2.ª columna de B.

R(1, P) = suma de los productos de los términos de la 1.ª fila de A por los de la P-ésima columna de B.

R(2, 1) = suma de los productos de los términos de la 2.ª fila de A por los de la 1.ª columna de B.

R(2, 2) = suma de los productos de los términos de la 2.ª fila de A por los de la 2.ª columna de B.

....

Ejemplo:

Sean

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 0 & 5 \end{pmatrix} \quad y \quad B = \begin{pmatrix} 6 & 3 \\ 1 & 0 \\ 2 & 5 \end{pmatrix}$$

$$R = A \times B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 0 & 5 \end{pmatrix} \begin{pmatrix} 6 & 3 \\ 1 & 0 \\ 2 & 5 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 * 6 + 2 * 1 + 3 * 2 & 1 * 3 + 2 * 0 + 3 * 5 \\ 4 * 6 + 0 * 1 + 5 * 2 & 4 * 3 + 0 * 0 + 5 * 5 \end{pmatrix} = \begin{pmatrix} 14 & 18 \\ 34 & 37 \end{pmatrix}$$

Al ser la dimensión de A, 2 × 3, y la de B, 3 × 2, la dimensión de P es 2 × 2 (el primer 2 corresponde al número de filas de A y el segundo 2 al número de columnas de B).

El programa

El programa que efectúe el producto de dos matrices tiene que leer sus dimensiones, dimensionar las matrices A, B y R e introducir los elementos de las matrices A y B.

```
1  REM PRODUCTO DE MATRICES
10 INPUT "FILAS DE LA PRIMERA MATRIZ"; M
20 INPUT "COLUMNAS DE LA PRIMERA MATRIZ O FILAS DE LA SE-
   GUNDA MATRIZ"; N
30 INPUT "COLUMNAS DE LA SEGUNDA MATRIZ"; P
40 DIM A(M, N)
50 DIM B(N, P)
60 DIM R(M, P)
```

En las instrucciones **10, 20** y **30** se introducen las dimensiones de las dos matrices, teniendo en cuenta que el número de columnas de la primera es igual al número de filas de la segunda. Las instrucciones **40, 50** y **60** dimensionan las matrices A, B y R.

```
70  PRINT "MATRIZ A"
80  FOR I = 1 TO M
90  PRINT "FILA"; I
100 FOR J = 1 TO N
110 PRINT "COLUMNA"; J
120 INPUT A(I, J)
130 NEXT J
140 NEXT I
150 CLS
```

Este bloque sirve para introducir los elementos de la matriz A, por filas. Se utilizan dos bucles: el de variable I para variar las filas, y el interno, de variable J, para variar las columnas (la fila I fija). Por último, la instrucción **150** borra la pantalla.

```
160 PRINT "MATRIZ B"
170 FOR I = 1 TO N
180 PRINT "FILA"; I
190 FOR J = 1 TO P
200 PRINT "COLUMNA"; J
210 INPUT B(I, J)
220 NEXT J
230 NEXT I
240 CLS
```

Este bloque es análogo al anterior; solamente se diferencia en que introduce los elementos de la matriz B.

```
250 FOR I = 1 TO M
260 FOR J = 1 TO P
270 FOR K = 1 TO N
280 R(I, J) = R(I, J) + A(I, K) * B(K, J)
290 NEXT K
300 NEXT J
310 NEXT I
```

El bucle **250-310,** de variable I, va fijando la fila de R y la de A.

El bucle **260-300,** de variable J, fija la columna de R y la de B.

El bucle **270-290,** de variable K, calcula el valor del elemento R(I, J) por medio de la instrucción **280.**

Por tanto, en este bloque se calculan los distintos elementos de la matriz producto R.

```
320 PRINT "LA MATRIZ PRODUCTO ES"
330 FOR I = 1 TO M
340 PRINT "FILA"; I
350 FOR J = 1 TO P
360 PRINT R(I, J)
370 NEXT J
380 NEXT I
390 END
```

Este grupo de instrucciones imprime los valores de las distintas filas de la matriz producto R. Con la instrucción **390** termina el programa.

Ejecución del programa

Si se ejecuta el programa dando como datos:

$$M = 2 \qquad N = 3 \qquad P = 2$$

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 0 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 6 & 3 \\ 1 & 0 \\ 2 & 5 \end{pmatrix}$$

el resultado que se obtiene en la pantalla es éste:

```
FILA 1
14
18
FILA 2
34
37
```

es decir:

$$R = \begin{pmatrix} 14 & 18 \\ 34 & 37 \end{pmatrix}$$

6. Resolución de sistemas de ecuaciones lineales

Presentación del programa

Vamos a analizar el caso de los sistemas lineales en los que el número de ecuaciones coincide con el número de incógnitas. Trataremos primero de determinar si tiene solución y, en caso afirmativo, hallarla.

Aunque el programa es válido para un sistema cualquiera de N ecuaciones con N incógnitas, estudiaremos el procedimiento refiriéndonos a un sistema de 3 ecuaciones con 3 incógnitas:

$$\left.\begin{array}{l} A(1, 1) * X(1) + A(1, 2) * X(2) + A(1, 3) * X(3) = A(1, 4) \\ A(2, 1) * X(1) + A(2, 2) * X(2) + A(2, 3) * X(3) = A(2, 4) \\ A(3, 1) * X(1) + A(3, 2) * X(2) + A(3, 3) * X(3) = A(3, 4) \end{array}\right\} \quad (1)$$

donde A(1, 1), ..., A(3, 3) son los coeficientes de las incógnitas X(1), X(2) y X(3), y A(1, 4), A(2, 4) y A(3, 4), los términos independientes.

Vamos a transformar este sistema en otro equivalente, reemplazando las ecuaciones 2.ª y 3.ª, y dejando fija la 1.ª Para ello realizamos las siguientes operaciones:

— Multiplicamos la 2.ª ecuación por A(1, 1) y la 1.ª por A(2, 1), y restamos ambas miembro a miembro. Obtenemos así una nueva ecuación, que reemplaza a la 2.ª, con el primer término nulo.

— Multiplicamos la 3.ª ecuación por A(1, 1) y la 1.ª por A(3, 1), y restamos ambas miembro a miembro. Obtenemos de este modo una nueva ecuación, que reemplaza a la 3.ª, con el primer término nulo.

Obtenemos así el siguiente sistema, equivalente al anterior:

$$\left.\begin{array}{r} A(1, 1) * X(1) + A'(1, 2) * X(2) + A(1, 3) * X(3) = A(1, 4) \\ A'(2, 2) * X(2) + A'(2, 3) * X(3) = A'(2, 4) \\ A'(3, 2) * X(2) + A'(3, 3) * X(3) = A'(3, 4) \end{array}\right\} \quad (2)$$

Transformamos este sistema en otro equivalente, dejando fijas la 1.ª ecuación y la 2.ª, y reemplazando la 3.ª Para ello multiplicamos la 3.ª ecuación por A'(2, 2) y la 2.ª por A'(3, 2), y restamos ambas miembro a miembro. Obtenemos así una nueva ecuación, que reemplaza a la 3.ª, con el primer y segundo términos nulos. Queda, entonces, el siguiente sistema equivalente:

$$\left.\begin{array}{r} A(1, 1) * X(1) + A(1, 2) * X(2) + A(1, 3) * X(3) = A(1, 4) \\ A'(2, 2) * X(2) + A'(2, 3) * X(3) = A'(2, 4) \\ A''(3, 3) * X(3) = A''(3, 4) \end{array}\right\} (3)$$

En este sistema, a partir de la 3.ª ecuación obtenemos el valor de X(3), dividiendo A''(3, 4) entre A''(3, 3). Reemplazando el valor de X(3) en la 2.ª ecuación obtenemos X(2) y, finalmente, reemplazando los valores de X(3) y X(2) en la 1.ª ecuación, hallamos el valor de X(1).

Este procedimiento es posible siempre que algún coeficiente de X(1) sea distinto de cero; en el ejemplo, estos coeficientes son A(1, 1), A(2, 1) y A(3, 1).

Si alguno de estos coeficientes es distinto de cero; el procedimiento conduce al sistema equivalente (2). En éste se repite el proceso con las ecuaciones del sistema, con exclusión de la primera. Así se llega al sistema equivalente (3).

Ahora bien, para que el sistema tenga solución es necesario que el coeficiente A(N, N) (en el ejemplo, A''(3, 3)) sea distinto de cero.

El programa

```
 10 PRINT "NUMERO DE ECUACIONES O INCOGNITAS"
 20 INPUT N
 30 DIM A(M, N + 1)
 40 FOR I = 1 TO N
 50 PRINT "ECUACION"; I
 60 FOR J = 1 TO N
 70 PRINT "COEFICIENTE"; J
 80 INPUT A(I, J)
 90 NEXT J
100 PRINT "TERMINO INDEPENDIENTE"
110 INPUT A(I, N + 1)
120 NEXT I
130 CLS
```

Esta parte del programa introduce en la memoria los coeficientes de las incógnitas y los términos independientes de las N ecuaciones.

La instrucción **20** introduce el número de ecuaciones del sistema.

```
140  FOR K = 1 TO N − 1
150  IF A(K, K) <> 0 THEN GOTO 250
160  FOR I = K + 1 TO N
170  IF A(I, K) <> 0 THEN GOTO 200
180  NEXT I
190  GOTO 430
```

Mediante estas instrucciones se analiza si algún coeficiente A(K, K) es distinto de cero y se busca uno que lo sea.

```
200  FOR M = K TO N + 1
210  B = A(I, M)
220  A(I, M) = A(K, M)
230  A(K, M) = B
240  NEXT M
```

Con las instrucciones **200** a **240** se permutan las ecuaciones (tener en cuenta que todavía no se ha cerrado el bucle de variable K).

Para el cambio de las ecuaciones se utiliza la variable B en la que se almacenan los resultados.

```
250  FOR I = K + 1 TO N
260  FOR J = K + 1 TO N + 1
270  A(I, J) = A(I, J) * A(K, K) − A(K, J) * A(I, K)
280  NEXT J
290  NEXT I
300  NEXT K
```

En esta parte del programa se obtienen sucesivos sistemas equivalentes hasta llegar a uno en el que figura una ecuación con una sola incógnita. Una vez conseguido este sistema se cierra el bucle de variable K.

```
310  IF A(N, N) = 0 THEN GOTO 430
320  PRINT "SISTEMA COMPATIBLE. LA SOLUCION ES:"
330  FOR K = N TO 2 STEP −1
340  B = A(K, N + 1)/A(K, K)
350  PRINT "X"; K; "="; B
360  FOR I = 1 TO K − 1
370  A(I, N + 1) = A(I, N + 1) − A(I, K) * B
380  NEXT I
390  NEXT K
400  X1 = A(1, N + 1)/A(1, 1)
410  PRINT "X 1 ="; X1
```

Mediante estas instrucciones se han ido obteniendo y sustituyendo el valor de las incógnitas hasta llegar al valor de X1.

```
420  GOTO 440
430  PRINT "EL SISTEMA NO ES COMPATIBLE DETERMINADO"
440  END
```

La instrucción **420** transfiere el control a la **440,** después de ser impreso el valor de la incógnita X1.

En el caso de que los coeficientes A(K, K) sean todos nulos, la instrucción **190** transfiere el control a la **430,** y si el coeficiente A(N, N) es nulo, la instrucción **310** también transfiere el control a la instrucción **430.**

Con la instrucción **440** se termina el programa.

Ejecución del programa

Si al ejecutar el programa se introduce la dimensión N = 4 y los coeficientes y términos independientes del sistema

$$\left.\begin{array}{r} 2x_1 - 1x_2 + 1x_3 - 2x_4 = -5 \\ 2x_1 + 2x_2 - 3x_3 + 1x_4 = -1 \\ -1x_1 + 1x_2 - 1x_3 + 0x_4 = -1 \\ 4x_1 - 3x_2 + 2x_3 - 3x_4 = -8 \end{array}\right\}$$

se obtiene en la pantalla el siguiente resultado:

```
SISTEMA COMPATIBLE. LA SOLUCION ES:
x 4 = 3
x 3 = 2
x 2 = 1
x 1 = 0
```

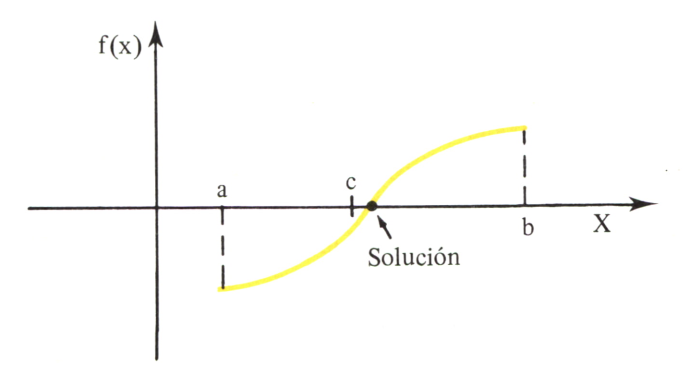
7. Resolución de una ecuación por el método del punto medio

Presentación del problema

Supongamos que f(x) es una función continua en el intervalo [a, b], y los signos de f(a) y de f(b) son distintos. Con estas condiciones, se puede demostrar que la ecuación

$$f(x) = 0$$

tiene solución en el intervalo [a, b]



El procedimiento de cálculo numérico que hay que seguir para obtener la solución con una precisión E determinada es el siguiente:

— se divide el intervalo por 2, obteniéndose así el punto medio c del intervalo [a, b] (de ahí el nombre del método);

— se verifica si la longitud del intervalo [a, b] es menor que E; es decir, se comprueba si $|a - b| < E$. Si es menor, la solución es c;

— se verifica si $f(c) > 0$; en caso afirmativo, se hace $b = c$ y en caso negativo se hace $a = c$. De esta forma se obtiene un nuevo intervalo [a, b] más pequeño que sigue cumpliendo las condiciones iniciales;

— se repite el proceso hasta lograr que c (solución) < E.

El programa

Para elaborar el programa hay que tener en cuenta que:

FNF(x) es la función que hay que definir en cada caso.
A y B son los extremos del intervalo. Estos valores deben cumplir la condición de que FNF(A) y FNF(B) tengan distinto signo.
E es la precisión con que se desea obtener la solución.
C es el punto medio de los sucesivos intervalos que se van obteniendo.

Aunque el programa que transcribimos empieza en la instrucción **20,** en cada caso concreto hay que añadir una instrucción **10** que defina la función correspondiente a la ecuación cuya solución se desee hallar.

```
20  INPUT "EXTREMOS DEL INTERVALO"; A, B
30  INPUT "PRECISION"; E
40  C = (A + B)/2
50  IF ABS(B − A) < E THEN PRINT "SOLUCION ="; C : END
60  IF FNF(C) > 0 THEN B = C ELSE A = C
70  GOTO 40
```

En las instrucciones **20** y **30** se introducen los valores de A, B y E.

La línea **40** calcula el punto medio C del segmento AB.

La línea **50** pregunta si el valor absoluto de la diferencia de A y B es menor que la precisión E; en caso afirmativo imprime la solución C y termina el proceso.

La línea **60** pregunta si la función en C es positiva; en caso afirmativo hace B = C y en caso contrario A = C.

La línea **70** envía el control a la **40** para que se repita el proceso.

Ejecución del programa

Si se desea hallar la solución de la ecuación

$$x^3 + x^2 - x - 1 = 0$$

previamente hay que definir la función

$$f(x) = x^3 + x^2 - x - 1$$

Luego la instrucción **10,** deberá ser ésta:

```
10  DEF FNF(x) = x ^ 3 + x ^ 2 − x − 1
```

Elegimos como extremo del intervalo a = 0 y b = 3.

Dado que

$$f(a) = f(0) = -1$$
$$f(b) = f(3) = 32$$

son de distinto signo, la solución se encuentra en el intervalo [0, 3].

Al ejecutar el programa y dar los datos:

$$A = 0$$
$$B = 3$$
$$E = 0.001$$

se obtiene el resultado que aparece en la pantalla.

```
EXTREMOS DEL INTERVALO? 0.3
PRECISION? 0.001
SOLUCION = 1.0001220703125
```

Observación

A este programa se le puede añadir una línea **25** para comprobar si la función tiene el mismo signo en los extremos A y B, en cuyo caso, imprime, un informe que expresa que estos extremos no valen y después envía el control a la instrucción **20**, para que se introduzcan nuevos valores de A y B.

```
25 IF FNF(A) * FNF(B) > = 0 THEN PRINT "NO VALEN" : GOTO 20
```

8. Resolución de la ecuación cúbica

Presentación del problema

La ecuación general de tercer grado es de la forma

$$Ax^3 + Bx^2 + Cx + D = 0$$

Para resolverla es necesario realizar una serie de cambios de variable. Los pasos que hay que dar son los siguientes:

1. Hacer este cambio de variable:

$$x = y - \frac{B}{3A}$$

$$A\left(Y - \frac{B}{3A}\right)^3 + B\left(Y - \frac{B}{3A}\right)^2 + C\left(Y - \frac{B}{3A}\right) + D = 0$$

Desarrollando y simplificando queda:

$$AY^3 + \left(- \frac{B^2}{3A} + C\right) Y + \left(\frac{2B^3}{27A^2} - \frac{BC}{3A} + D\right) = 0$$

Y dividiendo por A se obtiene:

$$Y^3 + \left(- \frac{B^2}{3A^2} + \frac{C}{A}\right) Y + \left(\frac{2B^3}{27A^3} - \frac{BC}{3A^2} + \frac{D}{A}\right) = 0$$

2. Haciendo

$$\begin{cases} 3P = - \dfrac{B^2}{3A^2} + \dfrac{C}{A} \; ; \; \boxed{\mathbf{P = - \dfrac{B^2}{9A^2} + \dfrac{C}{3A}}} \\ \boxed{\mathbf{Q = \dfrac{2B^3}{27A^3} - \dfrac{BC}{3A^2} + \dfrac{D}{A}}} \end{cases}$$

La ecuación anterior queda así:

$$\mathbf{Y^3 + 3PY + Q = 0}$$

3. Hacer este otro cambio de variable:

$$Y = U + V$$

Desarrollando

$$U^3 + 3U^2V + 3UV^2 + V^3 + 3P(U + V) + Q = 0$$

y simplificando queda:

$$U^3 + V^3 + (3UV + 3P)(U + V) + Q = 0$$

Entre U, V y P imponemos esta relación:

$$3UV + 3P = 0$$

de la que se deduce

$$UV = - P \text{ o su equivalente } V = \frac{-P}{U}$$

Sustituyendo V por $\frac{-P}{U}$ se obtiene la expresión

$$U^3 - \frac{P^3}{U^3} + Q = 0$$

o su equivalente

$$\boxed{\mathbf{U^6 + QU^3 - P^3 = 0}}$$

que es una ecuación cuadrática en U^3 ($(U^3)^2 + Q(U^3) - P^3 = 0$) cuya solución es:

$$U^3 = \frac{-Q \pm \sqrt{Q^2 + 4P^3}}{2}$$

Si se hubiera despejado V en función de U se habrían obtenido las mismas soluciones. En consecuencia se puede elegir:

$$U^3 = \frac{-Q + \sqrt{Q^2 + 4P^3}}{2} \quad \text{o} \quad V^3 = \frac{-Q - \sqrt{Q^2 + 4P^3}}{2}$$

Luego se obtienen tres soluciones para U y otras tres para V, que combinadas dan nueve soluciones, pero de ellas sólo se deben tomar las que cumplan la condición:

$$U \cdot V = -P$$

Con esta restricción, las tres raíces de la ecuación cúbica vienen dadas por la fórmula de Cardano:

$$\boxed{Y = \sqrt[3]{\frac{-Q + \sqrt{Q^2 + 4P^3}}{2}} + \sqrt[3]{\frac{-Q - \sqrt{Q^2 + 4P^3}}{2}}}$$

en la que el producto de las raíces cúbicas es igual a –P.

4. Para calcular las soluciones se tiene que tener en cuenta: el valor del discriminante de la ecuación $U^6 + QU^3 - P^3 = 0$

$$\boxed{T = Q^2 + 4P^3}$$

- Si $T < 0$ existen tres raíces reales:

$$u = \sqrt[3]{\frac{-Q \pm \sqrt{Q^2 + 4P^3}}{2}} = \sqrt[3]{-\frac{Q}{2} \pm i \sqrt{-\left(\frac{Q}{2}\right)^2 - P^3}} =$$

$$= \sqrt{L\,(\cos Z \pm i \operatorname{sen} Z)}$$

$$L = \sqrt[3]{\frac{Q^2}{4} - \frac{Q^2}{4} - P^3} = \sqrt{-P^3}$$

$$Z = \operatorname{arc\,tg} \frac{\sqrt{-\frac{Q^2}{4} - P^3}}{-\frac{Q}{2}}$$

$$U_1 = \sqrt[3]{L}\left(\cos\frac{Z}{3} + i\,\text{sen}\,\frac{Z}{3}\right)$$

$$U_2 = \sqrt[3]{L}\left(\cos\left(\frac{Z}{3} + \frac{2\pi}{3}\right) + i\,\text{sen}\left(\frac{Z}{3} + \frac{2\pi}{3}\right)\right)$$

$$U_3 = \sqrt[3]{L}\left(\cos\left(\frac{Z}{3} + \frac{4\pi}{3}\right) + i\,\text{sen}\left(\frac{Z}{3} + \frac{4\pi}{3}\right)\right)$$

$$V_1 = \sqrt[3]{L}\left(\cos\frac{Z}{3} - i\,\text{sen}\,\frac{Z}{3}\right)$$

$$V_2 = \sqrt[3]{L}\left(\cos\left(\frac{Z}{3} + \frac{2\pi}{3}\right) - i\,\text{sen}\left(\frac{Z}{3} + \frac{2\pi}{3}\right)\right)$$

$$V_3 = \sqrt[3]{L}\left(\cos\left(\frac{Z}{3} + \frac{4\pi}{3}\right) - i\,\text{sen}\left(\frac{Z}{3} + \frac{4\pi}{3}\right)\right)$$

Al imponer la condición U.V. = –P nos quedan las soluciones:

$$Y_1 = 2\sqrt[3]{L}\cos\frac{Z}{3} = 2\sqrt{-P}\cos\frac{Z}{3}$$

$$Y_2 = 2\sqrt[3]{L}\cos\left(\frac{Z}{3} + \frac{2\pi}{3}\right) = 2\sqrt{-P}\cos\left(\frac{Z}{3} + \frac{2\pi}{3}\right)$$

$$Y_3 = 2\sqrt[3]{L}\cos\left(\frac{Z}{3} + \frac{4\pi}{3}\right) = 2\sqrt{-P}\cos\left(\frac{Z}{3} + \frac{4\pi}{3}\right)$$

- Si T = 0 existen dos raíces reales, una de ellas doble.
  En este caso U = V, y las raíces son:
  $Y_1 = +2U$
  $Y_2 = -U$ (raíz doble)

- Si T > 0 existen una raíz real y dos complejas conjugadas.

  $Y_1 = U + V$ (raíz real)

  $$Y_2 = -\frac{U+V}{2} + \frac{\sqrt{3}}{2}(U - V)i$$

  $$Y_3 = -\frac{U+V}{2} - \frac{\sqrt{3}}{2}(U - V)i$$

5. Por último, hay que deshacer el cambio $X = Y - \frac{B}{3A}$, para obtener las soluciones de la ecuación $Ax^3 + Bx^2 + CX + D = 0$.

El programa

```
10   INPUT A, B, C, D
20   P = – B ↑ 2/(9 * A ↑ 2) + C/(3 * A)
30   Q = (2 * B ↑ 3) / (27 * A ↑ 3) – (B * C) / (3 * A ↑ 2) + D/A
40   T = Q ↑ 2 + 4 * P ↑ 3
50   IF T < 0 THEN GOSUB 1000 : END
60   U = (– Q + SQR(T))/2
70   IF U < 0 THEN U = –((–U) ↑ (1/3))
80   IF U > 0 THEN U = U ↑ (1/3)
90   IF T = 0 THEN GOSUB 2000 : END
100  V = (–Q – SQR(T))/2
110  IF V < 0 THEN V = – ((–V) ↑ (1/3))
120  IF V > 0 THEN V = V ↑ (1/3)
130  IF T > 0 THEN GOSUB 3000 : END
140  END
```

Éste es el bloque principal del programa, que llama a tres subrutinas, **1000, 2000** y **3000,** para resolver la ecuación según los valores del discriminante T.

— La línea **10** pide los coeficientes de la ecuación.

— Las líneas **20** y **30** calculan los valores de P y Q.

— La línea **40** calcula el valor del discriminante T.

— Si $T < 0$ la línea **50** envía el control a la subrutina **1000** para que encuentre las soluciones.

— Las líneas **60, 70** y **80** calculan el valor U. Distinguiendo entre los casos de $U < 0$, $U > 0$ y $U = 0$.

— Si $T = 0$ la línea **90** envía el control a la subrutina **2000** para que encuentre las soluciones.

— Las líneas **100, 110** y **120** calculan el valor de V.

— Si $T > 0$ entonces la línea **130** envía el control a la subrutina **3000** para que encuentre las soluciones.

```
1000  PRINT "TRES RAICES REALES"
1010  IF –0.01 < Q AND Q < 0.01 THEN Z = 1.5708 : GOTO 1030
1020  Z = ATN (SQR (–Q * Q/4 – P ↑ 3) / ABS (–Q/2))
1030  IF (–Q/2) < 0 THEN Z = 3.1416 – Z
1040  PRINT 2 * SQR(–P) * cos (Z/3) – B/(3 * A)
1050  PRINT 2 * SQR(–P) * cos (Z/3 + 6.2832/3) – B/(3 * A)
1060  PRINT 2 * SQR(–P) * cos (Z/3 + 2 * 6.2832/3) – B/(3 * A)
1070  RETURN
```

Esta subrutina se ejecuta cuando $T < 0$ y da como resultado tres raíces reales.

Las líneas **1010, 1020** y **1030** calculan el valor del argumento Z, averiguando si está en el primer cuadrante o en el segundo, o si es un ángulo recto (línea **1010**), en cuyo caso no se calcula en la línea **1020** para que no dé error (ya que no existe la tangente de un ángulo recto).

Las líneas **1040, 1050** y **1060** imprimen los resultados.

```
2000  PRINT "DOS RAICES REALES (UNA DOBLE)"
2010  PRINT 2 * U - B/(3 * A)
2020  PRINT - U - B/(3 * A); "RAIZ DOBLE"
2030  RETURN
```

Esta subrutina se ejecuta cuando T = 0, y da como resultado dos raíces reales, una de ellas doble.

```
3000  PRINT "UNA RAIZ REAL Y DOS COMPLEJAS"
3010  PRINT U + V - B/(3 * A)
3020  R = - (U + V)/2 - B/(3 * A)
3030  I = ABS (SQR(3)/2 * (U - V))
3040  PRINT R; "+"; I; "i"
3050  PRINT R; "-"; I; "i"
3060  RETURN
```

Esta subrutina se ejecuta cuando T > 0 y da como resultado una raíz real y dos complejas conjugadas.

La línea **3010** imprime la raíz real.

La línea **3020** calcula la parte real de las raíces complejas, y la **3030** la parte imaginaria en valor absoluto.

Las líneas **3040** y **3050** imprimen las raíces complejas.

Ejecuciones del programa

- La ecuación $X^3 - 6x^2 + 11x - 6 = 0$ tiene como soluciones x = 1; x = 2; x = 3 como muestra la ejecución del programa

```
run
? 1, -6, 11, -6
TRES RAICES REALES
2.99999929
1.00000031
1.99999733
OK
■
```

- La ecuación $x^3 + 3x^2 + 3x + 1 = 0$ tiene como soluciones $x = -1$; $x = -1$; $x = -1$. En la pantalla aparece una raíz doble:

```
run
? 1, 3, 3, 1
DOS RAICES REALES (UNA DOBLE)
–1
–1 RAIZ DOBLE
OK
■
```

- La ecuación $x^3 - 1 = 0$ tiene una solución real $x = 1$ y dos complejas $x = \frac{1}{2} \pm \frac{\sqrt{3}}{2}$ i. Al ejecutar el programa se obtiene:

```
run
? 1, 0, 0, –1
UNA RAIZ REAL Y DOS COMPLEJAS
1
–0.5 + 0.86602540 i
–0.5 – 0.86602540 i
OK
■
```

9. El calendario

1. Fechas de la Semana Santa de cualquier año

Presentación del problema

- La Semana Santa viene determinada por la fecha de la Pascua de Resurrección. La fecha de esta conmemoración religiosa es celebrada el primer domingo después del ***plenilunio*** (luna llena) que coincide o sigue inmediatamente al equinoccio de primavera (21 de marzo).

  El algoritmo que se expone a continuación permite determinar la fecha de la Pascua (día y mes), y es válido desde el año 1583, siguiente a la implantación del calendario gregoriano (15-10-1582).

  El calendario gregoriano acumula el error de un día al cabo de 3314 años, luego el equinoccio de primavera (21 de marzo) del año 1582 + 3314 = 4896 tendrá lugar el día 20 de marzo (un día antes).

  Por eso el algoritmo no es válido para calcular la Pascua de los años posteriores al 4896.

  Las fechas extremas de la Pascua son el 22 de marzo (cae en esta fecha si el 22 es domingo) y el 25 de abril (cae en esta fecha si el 20 de marzo es plenilunio).

  Luego la Pascua puede caer en cualquier día que se encuentre dentro de ambas fechas mencionadas (variación: 35 días).

- Algoritmo:

Dividendo	*Divisor*	*Cociente*	*Resto*
El año x	19	—	A
x	100	B	C
B	4	D	E
B + 8	25	F	—
B − F + 1	3	G	—
19A + B − D − G + 15	30	—	H
C	4	I	J
32 + 2E + 2I − H − J	7	—	K
A + 11H + 22K	451	L	—
H + K − 7L + 114	31	M	N

Como resultado se tiene: M, que da el mes y N + 1, que da el día de la Pascua el año x.

La Semana Santa será del lunes anterior, al domingo N + 1.

La Semana Santa del año x se extiende, entonces, desde el lunes anterior al domingo N + 1, hasta dicho domingo.

El programa

```
 10 INPUT "AÑO"; X
 20 IF X < 1583 OR X > = 1582 + 3314 THEN GOTO 10
 30 Z = X : Y = 19 : GOSUB 1000 : A = R
 40 Z = X : Y = 1000 : GOSUB 1000 : B = Q : C = R
 50 Z = B : Y = 4 : GOSUB 1000 : D = Q : E = R
 60 Z = B + 8 : Y = 25 : GOSUB 100 : F = Q
 70 Z = B - F + 1 : Y = 3 : GOSUB 1000 : G = Q
 80 Z = 19 * A + B - D - G + 15 : Y = 30 : GOSUB 1000 : H = R
 90 Z = C : Y = 4 : GOSUB 1000 : I = Q : J = R
100 Z = 32 + 2 * E + 2 * I - H - J : Y = 7 : GOSUB 1000 : K = R
110 Z = A + 11 + H + 22 * K : Y = 451 : GOSUB 1000 : L = Q
120 Z = H + K - 7 * L + 114 : Y = 31 : GOSUB 1000 : M = Q : N = R
```

En este conjunto de instrucciones se introduce el valor de año X, y realizan las divisiones que indica el algoritmo, llamando a la subrutina **1000** que calcula el cociente Q y el resto R de la división entera Z/Y.

```
130 DIM M$(12)
140 FOR I = 1 TO 12
150 READ M$(I)
160 DATA "ENERO", "FEBRERO", "MARZO", "ABRIL"
170 DATA "MAYO", "JUNIO", "JULIO", "AGOSTO"
180 DATA "SEPTIEMBRE", "OCTUBRE", "NOVIEMBRE", "DICIEM-
    BRE"
190 NEXT I
```

El bucle **140-190** asigna a la lista M$(I) los meses del año.

```
200 PRINT "LA PASCUA DEL AÑO"; X; "ES EL DIA"; N + 1; "DEL
    MES DE"; M$(M)
210 END
```

La línea **200** imprime el día y el mes en que cae el domingo de Pascua del año X.

```
1000  Q = INT(Z/Y)
1010  R = Z - Y * Q
1020  RETURN
```

La subrutina **1000** calcula el cociente Q y el resto R de la división entera de Z/Y.

Ejecuciones del programa

- Para averiguar las fechas de la Semana Santa del año 1986 basta con seguir los pasos que indica la pantalla

```
run
AÑO ? 1986
LA PASCUA DEL AÑO 1986 ES EL DIA 30
DEL MES DE MARZO
OK
■
```

  Luego la Semana Santa del año 1986 tendrá lugar del lunes 24 de marzo al domingo 30 de marzo.

- La Semana Santa del año 2000 caerá en las fechas que se obtienen a continuación

```
run
AÑO ? 2000
LA PASCUA DEL AÑO 2000 ES EL DIA 23
DEL MES DE ABRIL
OK
■
```

  Luego, la Semana Santa del año 2000 tendrá lugar del 17 al 23 de abril.

2. Día de la semana y día juliano, DJ

Presentación del problema

José Scaliger propuso, en 1582, una escala de tiempo continua, de 7980 años julianos de 365,25 días, que comienza el 1 de enero del año 4713 a. de C.

A esta escala de tiempo la llamó ***período juliano*** en honor de su padre Julio.

Para obtener el día juliano DJ de un año A de nuestro calendario basta con aplicar las siguientes reglas:

1. Se calcula el producto $(4712 + A) \cdot 365{,}25$.

2. Si el resultado es un número entero se le resta 1, y si es decimal, se toma su parte entera.

3. Si el año es siguiente al de la reforma gregoriana (4-10-1582) se tienen que realizar las siguientes operaciones a continuación del anterior:

   $1583 \leqslant A \leqslant 1700$: se resta 10 (*)

   $1701 \leqslant A \leqslant 1800$: se resta 11

   $1801 \leqslant A \leqslant 1900$: se resta 12

   $1901 \leqslant A \leqslant 2100$: se resta 13

   $2101 \leqslant A \leqslant 2200$: se resta 14, etc.

4. Se suma el resultado, el número de días transcurridos desde el comienzo del año A.

Para determinar el día de la semana basta con dividir por 7 el día juliano DJ, y según sea el resto, se harán estas asignaciones:

0 → LUNES

1 → MARTES

2 → MIERCOLES

3 → JUEVES

4 → VIERNES

5 → SABADO

6 → DOMINGO

(*) Se debe restar 10 cuando $A \geqslant 1583$, y también si $A = 1582$, siempre que se verifique $M > 10$ ó $M = 10$ y $D >= 15$.

El programa

```
10  INPUT "DIA, MES, AÑO"; D, M, A
20  Z = (4712 + A) * 365.25
30  IF Z = INT(Z) THEN Z = Z - 1 ELSE Z = INT(Z)
40  IF (A > = 1583 OR (A = 1582 AND (M > 10 OR (M = 10 AND
    D > = 15)))) AND A < = 1700 THEN Z = Z - 10
50  IF 1701 < = A AND A < = 1800 THEN Z = Z - 11
60  IF 1801 < = A AND A < = 1900 THEN Z = Z - 12
70  IF 1901 < = A AND A < = 2100 THEN Z = Z - 13
80  IF 2101 < = A AND A < = 2200 THEN Z = Z - 14
```

En este conjunto de instrucciones se introduce el día D, el mes M y el año A, y se calcula Z, que es el número de días transcurridos según el período juliano, hasta que comienza el año A.

```
 90  FOR I = 1 TO M
100  READ B
110  DATA 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334
120  NEXT I
```

Este bucle asigna a B es número de días de los meses anteriores a M.

```
130  IF (A/4 <> INT (A/4)) THEN GOTO 180
140  IF A < 1582 THEN GOTO 160
150  IF (A/100 = INT(A/100)) AND (A/400 <> INT(A/400)) THEN
     GOTO 180
160  IF M < = 2 THEN GOTO 180
170  B = B + 1
```

Las líneas **130-170** determinan si el año es bisiesto:

— La línea **130** comprueba si A no es divisible por 4, en cuyo caso el año no es bisiesto.

— La línea **140** comprueba si el año pertenece al período juliano.

— La línea **150** elimina los bisiestos seculares que no sean divisibles por **400,** según la reforma gregoriana.

— La línea **160** comprueba si el mes es enero o febrero, en cuyo caso no influye que el año sea bisiesto.

— La línea **170** añade un día al computo, por ser el año bisiesto.

```
180  Z = Z + B + D
190  Q = INT(Z/7)
200  R = Z - 7 * Q
```

La línea **180** calcula el día juliano Z y, la **200,** el resto de su división entre 7.

```
210  RESTORE 240
220  FOR I = 0 TO R
230  READ R$
240  DATA  "LUNES",  "MARTES",  "MIERCOLES",  "JUEVES",
     "VIERNES", "SABADO", "DOMINGO"
250  NEXT I
```

El bucle **220-250** lee el día de la semana R$ correspondiente al resto R.

```
260  PRINT "EL DIA"; D; "DEL MES"; M; "DEL AÑO"; A; "ES"; R$
270  PRINT "EL DIA JULIANO DJ CORRESPONDIENTE ES"; Z
280  END
```

La línea **260** imprime el día de la semana y la **270** el día juliano DJ.

Ejecuciones del programa

- ¿En qué día de la semana nació Carolina si la fecha de su nacimiento fue el 13 de abril de 1983?

```
run
DIA, MES, AÑO? 13, 4, 1983
EL DIA 13 DEL MES 4 DEL AÑO 1983
ES MIERCOLES
EL DIA JULIANO DJ CORRESPONDIENTE ES:
  2 4 4 5 4 3 8
OK
■
```

- ¿En qué día de la semana comenzará el año 2000?

```
run
DIA, MES, AÑO? 1, 1, 2000
EL DIA 1 DEL MES 1 DEL AÑO 2000 ES
SABADO
EL DIA JULIANO DJ CORRESPONDIENTE ES:
 2451545
OK
■
```

3. Determinación del número de días que hay entre dos fechas

Presentación del problema

El procedimiento que permite determinar el mismo número de días que hay entre dos fechas se reduce a calcular los días julianos correspondientes a las dos fechas y a restarlos. Sugerimos, en consecuencia, repasar los conceptos expuestos en el programa anterior.

El programa

```
10 INPUT "DIA, MES, AÑO"; D1, M1, A1
20 INPUT "DIA, MES, AÑO"; D2, M2, A2
30 D = D1 : M = M1 : A = A1 : GOSUB 100 : Z1 = Z
40 D = D2 : M = M2 : A = A2 : GOSUB 100 : Z2 = Z
50 PRINT "EL NUMERO DE DIAS ENTRE EL"; D1; ","; M1; ","; A1;
   "Y EL"; D2; ","; M2; ","; A2; "ES:"
60 PRINT ABS(Z2 - Z1)
70 PRINT "EL DIA JULIANO CORRESPONDIENTE AL"; D1; ","; M1;
   ","; "A1"; "ES:"; Z1
80 PRINT "EL DIA JULIANO CORRESPONDIENTE AL"; D2; ","; M2;
   ","; A2; "ES:"; Z2
90 END
```

Las líneas **10-90** constituyen el programa principal que permite introducir las dos fechas, llama a la subrutina **100** para que calcule los dos números julianos correspondientes, y por último, imprime el número de días entre las dos fechas y sus días julianos.

```
100  Z = (4712 + A) * 365.25
110  IF Z = INT(Z) THEN Z = Z – 1 ELSE Z = INT(Z)
120  IF (A > = 1583 OR (A = 1582 AND (M < 10 OR (M = 10 AND D
     < = 15)))) AND A < = 1700 THEN Z = Z – 10
130  IF 1701 < = A AND A < = 1800 THEN Z = Z – 11
140  IF 1801 < = A AND A < = 1900 THEN Z = Z – 12
150  IF 1901 < = A AND A < = 2100 THEN Z = Z – 13
160  IF 2101 < = A AND A < = 2200 THEN Z = Z – 14
```

En la línea **100** comienza la subrutina que calcula el número juliano Z. En el bloque de instrucciones **100-160** se calcula el número de días julianos que hay hasta el principio del año A.

```
170  RESTORE 200
180  FOR I = 1 TO M
190  READ B
200  DATA 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334
210  NEXT I
```

El bucle **180-210** calcula el número B de días transcurridos del año A antes del mes M.

```
220  IF (A/4 <> INT (A/4)) THEN GOTO 270
230  IF A < 1582 THEN GOTO 250
240  IF (A/100 = INT (A/100)) AND (A/400<>INT (A/400)) THEN
     GOTO 270
250  IF M < = 2 THEN GOTO 270
260  B = B + 1
270  Z = Z + B + D
280  RETURN
```

La línea **220** comprueba si el año no es bisiesto. La línea **230** comprueba si el año es anterior al de la reforma gregoriana del calendario (15-10-1582). La línea **240** elimina de los bisiestos los años seculares no divisibles por **400.** La línea **250** comprueba si el mes M es enero o febrero.

La línea **260** suma 1 al número de días transcurridos del año A si es bisiesto y si el mes M es distinto de enero o febrero.

La línea **270** calcula el día juliano Z.

Ejecución del programa

Si se desea saber el número de días que hay entre el 15 de febrero de 1963 y el 29 de julio de 1985 basta con ejecutar el programa introduciendo los datos, como muestra la pantalla:

```
run
DIA, MES, AÑO? 15, 2, 1963
DIA, MES, AÑO? 29, 7, 1985
EL NUMERO DE DIAS ENTRE EL 15, 2, 1963 y
EL 29, 7, 1985 ES:
8200
EL DIA JULIANO CORRESPONDIENTE AL
15, 2, 1863 ES: 2438076
EL DIA JULIANO CORRESPONDIENTE AL
29, 7, 1985 ES: 2446276
OK
■
```

Observación

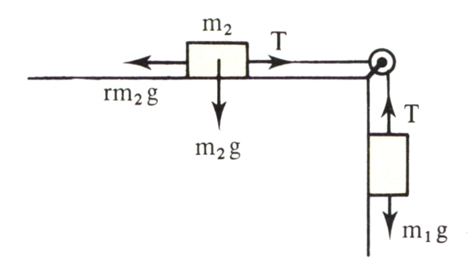
Una posible aplicación del programa es calcular el número de días vividos hasta la fecha. Para ello basta con introducir la fecha de nacimiento y la fecha de hoy.

10. Programas para estudiar conceptos de física y química

1. Cálculo de la aceleración de un sistema de dos masas

Presentación del problema

Sea el sistema de dos masas representado en esta figura.



Se trata de hallar la aceleración a que adquiere el sistema. Para ello aplicamos la ***segunda ley de Newton*** a las masas m1 y m2:

$$m_1\,g - T = m_1\,a$$
$$T - r\,m_2\,g = m_2\,a$$

siendo m_1 y m_2 los valores de las masas, T la tensión de la cuerda, r el coeficiente de rozamiento de la masa m_2 con la superficie horizontal, y $g = 9{,}8$ m/s^2.

Resolviendo el sistema, se obtiene el valor de la aceleración a.

$$a = \frac{(m_1 - r\,m_2) \cdot 9{,}8}{m_1 + m_2}$$

Si el rozamiento no es elevado, la aceleración a puede ser positiva, con lo que el sistema caería de forma acelerada.

Si, en cambio, el rozamiento es lo suficientemente elevado pueden presentarse estas situaciones:

a) Que $a < 0$ $a = 0$ y v (velocidad inicial) sea nula. En este caso, el sistema no se movería.

b) Que $a = 0$ y $v = 0$. En este caso, el sistema se movería con velocidad constante.

El programa

El programa resuelve el problema planteado, teniendo en cuenta los distintos casos que se pueden presentar.

```
 1 REM ACELERACION
10 PRINT "MASA 1"
20 INPUT M1
30 PRINT "MASA 2"
40 INPUT M2
50 PRINT "COEFICIENTE DE ROZAMIENTO"
60 INPUT R
70 PRINT "VELOCIDAD INICIAL"
80 INPUT V
```

En estas instrucciones se introducen los datos para resolver el problema.

```
90 A = (M1 - R * M2) * 9.8 / (M1 + M2)
```

En la instrucción **90** se calcula el valor de la aceleración.

```
100 IF A = 0 THEN GOTO 140
110 IF A < 0 THEN GOTO 160
120 PRINT "LA ACELERACION ES"; A; "M/(S ↑ 2)"
130 GOTO 170
140 IF V = 0 THEN GOTO 160
150 PRINT "LA VELOCIDAD ES SIEMPRE IGUAL A"; V; "M/S"
160 PRINT "EL SISTEMA NO SE MUEVE"
170 END
```

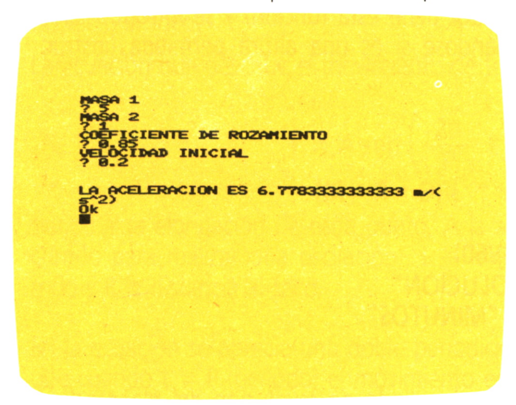
En las instrucciones **100-110** se averigua si la aceleración es positiva; en caso afirmativo se imprime su valor (instrucción **120**) y, en caso contrario, se tienen en cuenta las distintas alternativas.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introducen estos valores

M1 = 5 kg
M2 = 1 kg
R = 0.85
V = 0.2 m/s

se obtiene el resultado que muestra la pantalla:



2. Cálculo del período de revolución de un satélite

Presentación del problema

El período T de revolución de un satélite en órbita alrededor de la Tierra viene expresado por la fórmula

$$T = 2\pi \sqrt{\frac{(R + A)^3}{g\, R^2}}$$

siendo R el radio de la Tierra, que se supone igual a 6370000 metros, y la aceleración de la gravedad (9,8 m/s^2, aproximadamente) y A la altura sobre la superficie terrestre. (Para que la fórmula dé resultados correctos A debe ser mayor que 200 km y menor que 50 000 km.)

El programa

El programa calcula el período T e imprime en la pantalla el período de revolución en horas y minutos.

```
 1 REM PERIODO DE REVOLUCION
10 G = 9.8
20 R = 6370000
30 PI = 3.14159
40 INPUT "ALTURA DEL SATELITE EN KM"
60 IF A > 200 AND A < 50000 THEN GOTO 90
70 PRINT "DATO INCORRECTO"
80 GOTO 40
```

En estas instrucciones se asignan valores a las variables G. R y PI (no es necesario definir el valor de PI si el microordenador tiene esta función) y se introduce después el valor de la altura, comprobándose si es una altura permitida (instrucción **60**).

```
90  A = A * 1000
100 T = 2 * PI * SQR ((R + A) ↑ 3/(G * R * R))
110 H = INT (T/3600)
120 MN = INT ((T − 3600 * H)/60)
130 PRINT "PERIODO DE REVOLUCION"
140 PRINT H; "HORAS"; MN; "MINUTOS"
```

En esta parte del programa se calcula el período de revolución del planeta (instrucción **100**). Previamente, en la instrucción **90** se expresa la altura en metros. En las instrucciones **110** y **120** se expresa el período de revolución en horas y minutos, pues en la instrucción **100** está expresado en segundos.

```
150 PRINT "HAY OTRA ALTURA"
160 INPUT S$
170 IF S$ = "SI" THEN 40
180 END
```

Finalmente se pregunta si se va a introducir una nueva altura y, en caso afirmativo, se empieza de nuevo el cálculo.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introduce 2300 para A, se obtiene el valor 2 horas 14 minutos para T.

3. Análisis de las ondas de una cuerda

Presentación del problema

Cuando en un medio lineal, como por ejemplo una cuerda, se establece en un punto (el foco) una vibración armónica, ésta se propaga por la cuerda y cada

punto de ella vibra con un movimiento vibratorio armónico pero desfasado con respecto al foco.

La ecuación que describe el movimiento de todos los puntos de la cuerda es:

$$Y = A \operatorname{sen}\left(\frac{2\pi}{P}\ t - \frac{2\pi}{L}\ R\right)$$

siendo Y la elongación de cada punto, A la amplitud, P el período (tiempo transcurrido para realizar una oscilación), L la longitud de onda, t el tiempo transcurrido y R la distancia al foco.

En la ecuación se aprecia una doble periodicidad: una con respecto al tiempo, ya que cuando t = P (período) el movimiento en un punto se repite, y otra con respecto a la distancia, ya que los puntos situados a 1 L, 2 L ... vibran en fase.

A esta distancia L se llama longitud de onda.

Se llama frecuencia F al número de oscilaciones que experimenta un punto del medio por unidad de tiempo, verificándose que **F = 1/P.**

Por otro lado, existe una relación entre la longitud de onda y el período, y es la siguiente:

$$L = v\,T = V/F$$

en la que v es la velocidad de propagación de la perturbación (que depende de las características del medio y no del movimiento del foco).

El programa

El programa siguiente analiza la doble periodicidad respecto al tiempo y a la distancia.

```
1  REM ONDAS
10 INPUT "AMPLITUD"; A
20 INPUT "FRECUENCIA"; F
30 INPUT "VELOCIDAD"; V
40 L = V/F : P = 1/F
50 DEF FN F(T, X) = A * SIN (2 * 3.14 * (T/P - X/L))
```

En este conjunto de instrucciones se introducen los datos y se calculan la longitud de onda y el período.

En la instrucción **50** se define la función que describe a la onda.

```
60  T1 = 0 : T 2 = 3 * P : Y1 = –A : Y2 = A
70  DT = (T2 – T1) / 245 : DY = (Y2 – Y1) / 180
```

La instrucción **60** determina los valores máximo y mínimo del tiempo t y de la elongación y.

La instrucción **70** determina el valor de las escalas que se van a utilizar en la representación gráfica.

```
 80  SCREEN 2
 90  FOR X = 0 TO 2 * L STEP L/2
100  LINE (0, 90) – (255, 90)
110  LINE (15, 0) – (15, 180)
120  FOR T = 0 TO 3 * P STEP DT
130  Y = FN F(T, X)
140  X3 = INT ((T – 0) / DT + 0.5)
150  Y3 = INT ((Y – Y1) / DY + 0.5)
160  PSET (15 + X3, 180 – Y3)
170  NEXT T
180  CLS
190  NEXT X
200  GOTO 200
210  END
```

Este grupo de instrucciones dibuja las gráficas de los movimientos vibratorios de puntos situados a distancias X = 0, L/2, L, L + L/2 y 2L del foco. Para ello se utiliza el bucle **90-190.** Cada uno de estos movimientos vibratorios se representa para el tiempo transcurrido entre 0 y 3 * P (bucle **120-170**).

Ejecución del programa

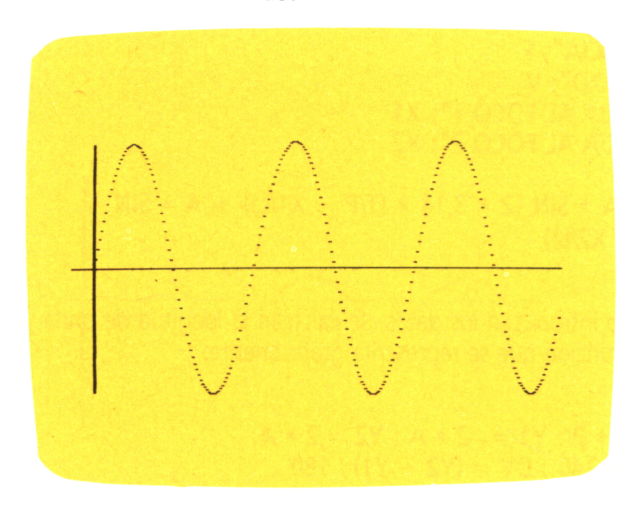
Si después de teclear **RUN** y RETURN se introducen estos valores

A = 0.5 m

F = 800 Hz

V = 1200 m s^{-1}

se obtiene este resultado:



En las gráficas de esta pantalla se comprueba que para cada longitud de onda L se repite el mismo movimiento.

4. Estudio de las interferencias

Presentación del problema

Un punto de un medio, por ejemplo de una cuerda, cuando es alcanzado simultáneamente por dos movimientos, vibra con un movimiento que es suma del que producirían cada uno de ellos independientemente.

Consideremos el caso de dos movimientos ondulatorios de la misma amplitud y frecuencia.

La interferencia se produce debido a la diferencia de caminos recorridos.

La ecuación que representa el movimiento de cada punto de la cuerda puede obtenerse sumando las funciones de onda de cada uno de los movimientos:

$$Y = Y1 + Y2 = A \text{ sen } (2\pi(T/P - X1/L)) + A \text{ sen } (2\pi(T/P - X2/L))$$

El programa

Mediante el siguiente programa se demuestra que si X2 − X1 = KL (K = 1, 2, 3, ...), siendo L la longitud de onda, el efecto de la interferencia corresponde a un máximo y si X2 − X1 = (2K − 1) L/2 (K = 1, 2, 3, ...) se produce una interferencia destructiva.

```
1  REM INTERFERENCIAS
10 CLS
20 INPUT "FRECUENCIA"; F
30 INPUT "VELOCIDAD"; V
40 INPUT "DISTANCIA AL FOCO 1"; X1
50 INPUT "DISTANCIA AL FOCO 2"; X2
60 L = V/F : P =1/F
70 DEF FN F(T) = A * SIN (2 * 3.14 * (T/P - X1/L)) + A * SIN
   (2 * 3.14 * (T/P - X2/L))
```

Con estas instrucciones se introducen los datos, se calculan la longitud de onda y el período y se define la función que se representa gráficamente.

```
80 T1 = 0 : T2 = 3 * P : Y1 = -2 * A : Y2 = 2 * A
90 DT = (T2 - T1) / 240 : DY = (Y2 - Y1) / 180
```

Mediante estas instrucciones se determinan los valores máximo y mínimo del tiempo t y de la elongación y.

```
100 SCREEN 2
110 LINE (0, 90) - (250, 90)
120 LINE (15, 0) - (15, 180)
130 FOR T = 0 TO 3 * P STEP DT
140 Y = FN F(T)
150 X3 = INT ((T - 0) / DT + 0.5)
160 Y3 = INT ((Y - Y1) / DY + 0.5)
170 PSET (15 + X3, 180 - Y3)
180 NEXT T
200 GOTO 200
210 END
```

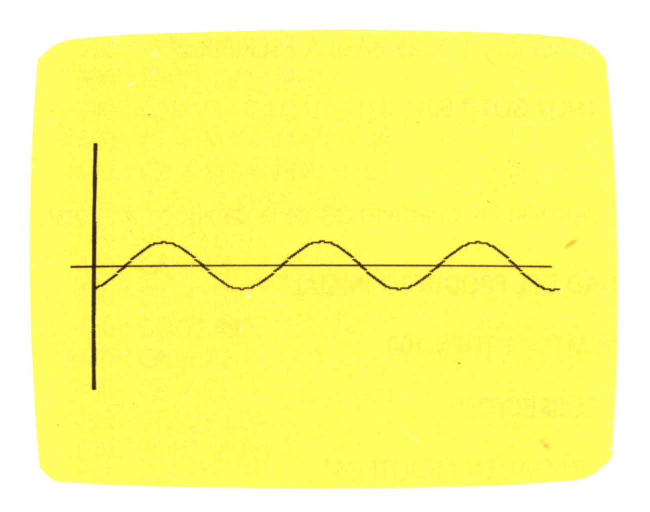
Las instrucciones **110** y **120** dibujan los ejes coordenados. Las restantes instrucciones dibujan la gráfica de la elongación e respecto al tiempo.

Ejecución del programa

Después de teclear **RUN** y RETURN introducir, por ejemplo, estos valores:

A = 0.5 m
F = 50 HZ
V = 340 m/s

Se obtiene, entonces, una longitud de onda de 6.8 m, y según el valor de X2 − X1 aparecerán máximos y mínimos.



5. Valoración ácido fuerte-base fuerte

Presentación del problema

La ecuación (no simplificada) que expresa el equilibrio ácido fuerte-base fuerte viene dada por la expresión

$$x^2 + (C_B - C_A)\,x - K_E = 0$$

siendo x la concentración molar de iones H_2O^+ presentes en la disolución, C_B y C_A las concentraciones molares de la base y el ácido en cada etapa de disolución, y K_B el producto iónico del agua, que depende de la temperatura.

Se trata de estudiar la variación del pH en el curso de una valoración de un ácido fuerte, por ejemplo, ácido clorhídrico con una base fuerte, como puede ser hidróxido de sodio, o a la inversa.

De las dos raíces de la ecuación sólo interesa la positiva. Sin embargo, debido al pequeño valor de K_B (10^{-14} a 25 °C) y al número de cifras significativas que maneja un microordenador, podría encontrarse que la raíz cuadrada llegara a tomar el valor cero, dando una solución de x negativa; ésta, al calcular el logaritmo ($pH = -\log [H_3O^+]$), produciría error, dado que no existen logaritmos decimales de números negativos.

Para solucionar este problema, calculamos la mayor de las dos raíces en valor absoluto. Si es positiva, es la que buscamos, y si es negativa, calculamos el producto de las raíces ($x1 \cdot x2 = c/a = K_E$) para obtener la otra.

El programa

```
10 PRINT "CARACTERISTICAS DE LA DISOLUCION A VALORAR"
20 PRINT "SI ES ACIDA ESCRIBIR 1, SI ES BASICA ESCRIBIR 2"
30 INPUT P
40 IF P = 1 OR P = 2 THEN GOTO 60
50 GOTO 30
```

En estas instrucciones se determinan las características de la disolución a valorar.

```
 60 PRINT "MOLARIDAD DEL PRODUCTO INICIAL"
 70 INPUT M1
 80 IF M1 < 0.001 OR M1 > 1 THEN 100
 90 GOTO 120
100 PRINT "DATOS INCORRECTOS"
110 GOTO 70
120 PRINT "VOLUMEN INICIAL EN MILILITROS"
130 INPUT V1
140 IF V1 < 0 OR V1 > 100 THEN 160
150 GOTO 180
160 PRINT "DATO INCORRECTO"
170 GOTO 130
180 PRINT "DISOLUCION UTILIZADA EN LA VALORACION"
190 PRINT "MOLARIDAD"
200 INPUT M2
210 IF M2 < 0.001 OR M2 > 1 THEN 230
220 GOTO 250
230 PRINT "DATO INCORRECTO"
240 GOTO 200
250 PRINT "VOLUMEN TOTAL VERTIDO EN MILILITROS"
260 INPUT VT
270 IF VT < = 1 OR VT > 10 * V1 THEN 290
280 GOTO 310
290 PRINT "DATO INCORRECTO"
300 GOTO 260
310 PRINT "CUANTOS MILILITROS SE AÑADEN CADA VEZ"
320 INPUT IV
330 IF IV < VT/20 OR IV > VT/5 THEN 350
340 GOTO 370
350 PRINT "DATO INCORRECTO"
360 GOTO 320
```

Hasta aquí se han introducido los datos necesarios verificando su corrección.

En las instrucciones **80, 140, 270, 330** se han establecido las condiciones habituales en un laboratorio.

```
370  PRINT
380  PRINT "V", "PH"
390  FOR V = 0 TO VT STEP IV
400  C1 = M1 * V1/V1 + V)
410  C2 = M2 * V/(V1 + V)
420  IF P = 1 THEN 440
430  GOTO 470
440  CA = C1
450  CB = C2
460  GOTO 490
470  CA = C2
480  CB = C1
490  GOSUB 600
500  PRINT V, PH
510  NEXT V
520  STOP
```

Con este grupo de instrucciones se calculan las concentraciones de las sustancias, teniendo en cuenta que sean ácido o base, y se envía el control del programa (cada vez que es necesario) a la subrutina **600** para calcular el pH.

```
600  KE = 1E - 14
610  D = SQR ((CB - CA) ↑ 2 + 4 * KE)
620  H1 = (- (CB - CA) + D) / 2
630  H2 = (- (CB - CA) - D) / 2
640  IF (CB - CA) < 0 THEN 670
650  H = -KE/H2
660  GOTO 680
670  H = H1
680  PH = - LOG (H) / 2.3
690  RETURN
700  END
```

Mediante esta subrutina se calculan las soluciones de la ecuación, teniendo en cuenta las indicaciones que se hicieron previamente.

Posteriormente se calcula el pH **(pH = − log [$H_3 0^-$] = − LOG [H] / 2.3)**.

En la instrucción **680 LET PH = − LOG (H) / 2.3,** la función LOG (X) calcula el logaritmo neperiano, de ahí que haya que dividir por la constante 2.3 para obtener el logaritmo decimal (*).

(*) Ver libro.

Ejecución del programa

Después de teclear **RUN** y RETURN introducir, por ejemplo, estos datos:

P = 1

M1 = 0.05

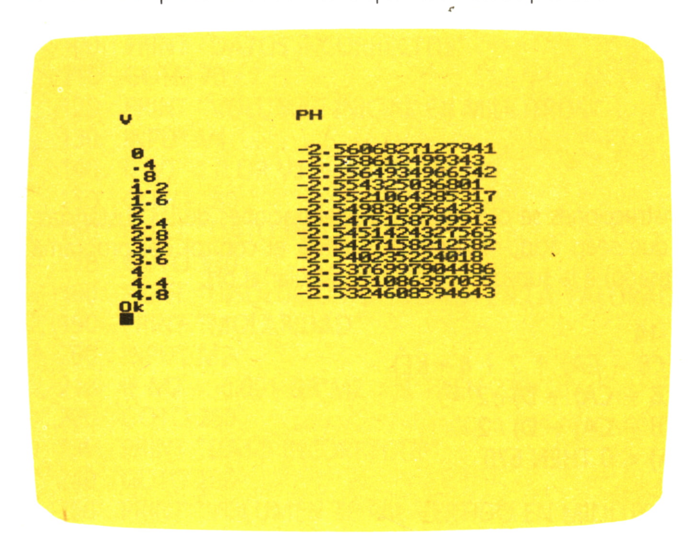
V1 = 85

M2 = 0.1

VT = 5

IV = 0.4

El resultado que se obtiene es el que muestra la pantalla.



6. Gráficas de puntos de las temperaturas de fusión, de ebullición y de las energías de ionización de los 40 primeros elementos del sistema periódico

Presentación del problema

La temperatura de fusión y de ebullición, así como la energía de ionización de los elementos son propiedades periódicas cuya variación depende de su número atómico.

La variación de estas propiedades periódicas se puede apreciar representándolas gráficamente.

El programa

El siguiente programa representa gráficamente:

— el punto de fusión en función del número atómico;
— el punto de ebullición en función del número atómico, y
— la energía de ionización en función del número atómico.

```
10  DIM Y (40), M (40), B (40), E (40), H (40), V (40), X (40)
20  FOR I = 1 TO 40
30  READ M (I), B (I), E (I)
40  NEXT I
50  PRINT "ESCRIBE:"
60  PRINT : PRINT "1 – GRAFICA PUNTO DE FUSION – NUMERO
    ATOMICO"
70  PRINT "2 – GRAFICA PUNTO DE EBULLICION – NUMERO ATO-
    MICO"
80  PRINT "3 – GRAFICA ENERGIA DE IONIZACION – NUMERO ATO-
    MICO"
90  PRINT "SI QUIERES SALIR DE UNA GRAFICA"
100 PRINT "PULSA UNA TECLA"
110 INPUT A
120 IF A < 1 OR A > 3 THEN GOTO 50
130 ON A GOTO 1000, 2000, 3000
```

Este conjunto de instrucciones lee los datos correspondientes a los distintos elementos y presenta las opciones que dispone el programa.

```
200 REM DIBUJA
210 CLS
220 X1 = 0 : X2 = 40
230 Y2 = Y (1) : Y1 = Y(1)
240 FOR I = 2 TO 40
250 IF Y2 < Y (I) THEN Y2 = Y (I)
260 IF Y1 > Y (I) THEN Y1 = Y (I)
265 X(I) = I
270 NEXT I
280 IF Y1 > 0 THEN Y1 = 0
290 DX = 40 : DY = Y2 – Y1
300 XS = 245/DX : YX = 175/DY
```

Estas instrucciones permiten escoger los valores máximo y mínimo de la variable Y (I), así como establecer las escalas de acuerdo a las cuales se realizará el dibujo.

```
310  SCREEN 2
320  FOR I = 1 TO 40
330  H (I) = X (I) * XS : V (I) = (Y (I) + ABS (Y1)) * YS
340  NEXT I
350  PSET (H (1), 175 − V (1))
360  FOR I = 2 TO 40
370  LINE (H (I − 1), 175 − V (I − 1)) − (H (I), 175 − V (I))
375  NEXT I
380  A$ = INKEY$
390  IF A$ = "" THEN GOTO 380
400  CLS : SCREEN 0 : GOTO 50
```

Este conjunto de instrucciones representa la gráfica Y (I) − I en la pantalla del microordenador, uniendo con tramos rectos los distintos puntos.

```
1000  FOR I = 1 TO 40
1010  Y (I) = M (I)
1020  NEXT I
1030  GOTO 200
2000  FOR I = 1 TO 40
2010  Y (I) = B (I)
2020  NEXT I
2030  GOTO 200
3000  FOR I = 1 TO 40
3010  Y (I) = E (I)
3020  NEXT I
3030  GOTO 200
```

En estos bloques de instrucciones la variable Y (I) adquiere el valor de M (I), B (I) o E (I), según la opción.

```
4000  REM DATOS
4001  DATA –259, –253, 13.6
4002  DATA –270, –269, 24.6
4003  DATA 180, 1360, 5.4
4004  DATA 1285, 2470, 9.3
4005  DATA 2030, 3700, 8.3
4006  DATA ...........................
........
```

Estas instrucciones **DATA** contienen sucesivamente los puntos de fusión, ebullición y energía de ionización de los primeros 40 elementos.

Ejecución del programa

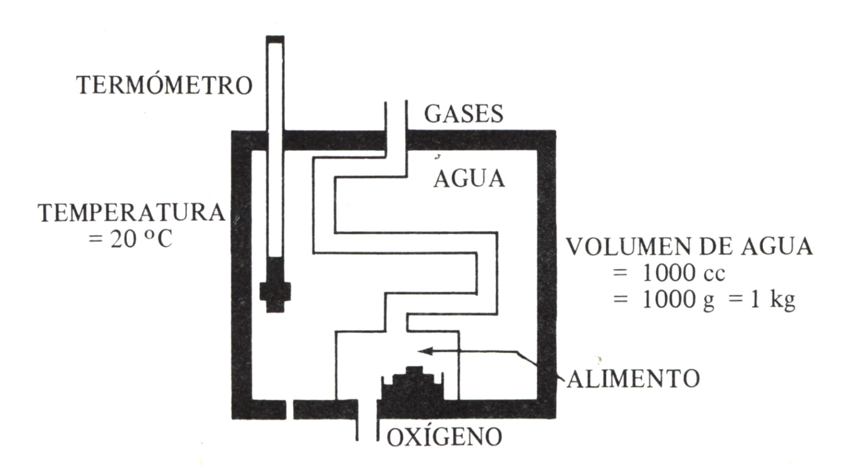
Si después de teclear **RUN** y | RETURN | se introduce el valor 3 para A, se obtiene la gráfica de la energía de ionización en función del número atómico.



7. Valores energéticos de los alimentos

Presentación del problema

La determinación del valor energético de un alimento se consigue quemando en un crisol una determinada cantidad del mismo dentro de una cámara rodeada de agua y por la que pasa una corriente de oxígeno.



Los gases calientes procedentes de la combustión pasan a través de un serpentín rodeado de agua, lo que hace que la temperatura del agua se eleve.

El valor energético se calcula mediante la fórmula

$$\triangle Q = m \triangle t \cdot 0{,}42$$

en la que $\triangle Q$ es el contenido energético del alimento, m su masa en gramos, $\triangle t$ la variación de temperatura (siendo la temperatura inicial 20 °C) y 4.2 el equivalente mecánico del calor (1 caloría = 4.18 julios).

El programa

```
 10 CLS
 20 PRINT "EL PESO PUEDE VARIAR ENTRE 0.5 y 3.0 GRAMOS"
 30 PRINT "PULSA UNA TECLA PARA CONTINUAR"
 40 A$ = INKEY$
 50 IF A$ = "" THEN GOTO 40
100 INPUT "TECLEA EL CODIGO NUMERICO PARA EL ALIMENTO"; F
110 RESTORE
120 FOR I = 1 TO F
130 READ F$
140 READ J
150 NEXT I
160 PRINT F$
```

La instrucción **100** pide el código numérico del alimento.

```
200 INPUT "ESCRIBE LA MASA DEL ALIMENTO EN GRAMOS
    (0.5 – 3.0)"; M
210 T1 = J * M/(4.2 * 1000) + 20
220 T = INT (T1 * 10 + 0.5)/10
230 PRINT "T ="; T
240 CLS
250 PRINT "PULSA UNA TECLA PARA CONTINUAR"
260 A$ = INKEY$
270 IF A$ = "" THEN GOTO 260
280 PRINT "LISTA DE DATOS"
290 PRINT : PRINT "ALIMENTO:"; F$
300 PRINT "MASA DE LA COMIDA ="; M; "G"
310 PRINT "VOLUMEN DE AGUA = 1000 cc"
320 PRINT "MASA DE AGUA = 1000 G"
330 PRINT "TEMPERATURA INICIAL = 20 C"
340 PRINT "TEMPERATURA FINAL = "; T
350 PRINT "CAMBIO EN LA TEMPERATURA ="; T – 20
```

Con estas instrucciones se calcula la temperatura final y se imprimen los resultados.

```
360  PRINT "PULSA UNA TECLA PARA CONTINUAR"
370  A$ = INKEY$
380  IF A$ = "" THEN GOTO 370
390  CLS
400  PRINT "1 GRAMO DE AGUA REQUIERE 4.2 JULIOS (J) DE ENER-
     GIA"
410  PRINT "PARA ELEVAR SU TEMPERATURA UN GRADO CENTI-
     GRADO"
420  PRINT : PRINT "VALOR ENERGETICO (J) ="
430  PRINT : PRINT "4.2 * ELEVACION DE TEMPERATURA (C) *
     MASA DE AGUA (G)"; "/"; "MASA DE ALIMENTO (G)"
440  PRINT : PRINT "EL VALOR ENERGETICO (EN JULIOS)"; F$; "=";
     J; "J"
450  PRINT "EL VALOR ENERGETICO (EN KJ) DE "; F$;" ="; J/1000;
     "KJ"
460  PRINT "PULSA UNA TECLA PARA CONTINUAR"
470  A$ = INKEY$
480  IF A$ = "" THEN GOTO 470
490  CLS
500  PRINT "PULSA E PARA EMPEZAR"
510  PRINT "PULSA T Y RETURN PARA ACABAR"
520  INPUT I$
530  IF I$ = "E" THEN GOTO 10
540  IF I$ = "T" THEN END
1000 DATA MANZANAS, 2000
1010 DATA BACON, 20000
1020 DATA MANTEQUILLA, 38000
1030 DATA ZANAHORIAS, 800
1040 DATA QUESO, 20000
1050 DATA POLLO, 6000
........
```

La instrucción **1000** y las siguientes almacenan los nombres de los alimentos y el valor energético de una masa fija de ellos.

A cada uno de estos alimentos se le asocia un código numérico que corresponde al lugar que ocupan en las instrucciones **DATA.**

Así, en el orden arbitrario supuesto a las manzanas les correspondería el código 1, al bacon el 2, a la mantequilla el 3, y así sucesivamente. Este código es importante para localizar el elemento cuando se lee.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introduce el valor 3 para el código F, en la pantalla se obtiene el valor energético de la mantequilla.

11. Dos programas para utilizar en el laboratorio

1. Pérdida de masa de un sólido

El programa siguiente indica los pasos que deben seguirse en el laboratorio para realizar un experimento, registra los resultados y realiza las operaciones matemáticas necesarias para obtener conclusiones.

```
  1 REM LABORATORIO
 10 PRINT "PERDIDA DE MASA DE UN SOLIDO CUANDO SE CA-
    LIENTA"
 20 PRINT : PRINT "PULSA UNA TECLA PARA CONTINUAR"
 30 A$ = INKEY$
 40 IF A$ = "" THEN GOTO 30
 50 CLS
 60 PRINT "INSTRUCCIONES"
 70 FOR K = 1 TO 1000 : NEXT K
 80 PRINT : PRINT "1 – COGE UN CRISOL VACIO"
 90 FOR K = 1 TO 1000 : NEXT K
100 PRINT : PRINT "2 – PESA EL CRISOL VACIO"
110 FOR K = 1 TO 1000 : NEXT K
120 PRINT : PRINT "3 – PESA EL CRISOL + EL SOLIDO"
130 FOR K = 1 TO 1000 : NEXT K
140 PRINT : PRINT "4 – CALIENTA HASTA QUE NO HAYA CAM-
    BIO..."
150 FOR K = 1 TO 1000: NEXT K
160 PRINT : PRINT "5 – DEJA ENFRIAR..."
170 FOR K = 1 TO 1000 : NEXT K
180 PRINT : PRINT "6 – PESA EL RECIPIENTE + RESIDUO"
190 FOR K = 1 TO 1000 : NEXT K
200 PRINT : PRINT "7 – REGISTRA TODOS LOS RESULTADOS"
```

En este conjunto de instrucciones se dan las indicaciones necesarias para realizar el experimento. Después de cada indicación se introduce un tiempo de espera con la ayuda de un bucle.

```
205 PRINT : PRINT : PRINT
210 PRINT "PULSA UNA TECLA PARA CONTINUAR"
220 A$ = INKEY$
230 IF A$ = "" THEN GOTO 220
240 CLS
300 PRINT "MEDIDAS" : PRINT
310 INPUT "CUAL ES LA MASA DEL CRISOL VACIO"; A
320 IF A < = 0 THEN PRINT "EL CRISOL DEBE PESAR ALGO" : FOR
    K = 1 TO 1000 : NEXT K : GOTO 310
330 CLS
340 INPUT "CUAL ES LA MASA DEL CRISOL + SOLIDO"; B
350 A1 = A
360 IF B < = A1 THEN PRINT "EL CRISOL + SOLIDO DEBEN SER
    MAS PESADOS : FOR K = 1 TO 1000 : NEXT K : GOTO 310
370 CLS
400 PRINT "CALIENTA HASTA QUE NO HAYA CAMBIOS..."
410 FOR K = 1 TO 1000 : NEXT K
420 PRINT "Y DEJA ENFRIAR"
430 FOR K = 1 TO 1000 : NEXT K
440 CLS
450 PRINT "CUAL ES LA MASA DEL" : INPUT "CRISOL + RESIDUO"; C
460 IF C > = B THEN PRINT "EL SOLIDO DEBE SER MAS LIGERO
    ANTES QUE DESPUES DE CALENTAR" : FOR K = 1 TO 1000 :
    NEXT K : GOTO 450
470 IF C < = A1 THEN PRINT "EL CRISOL Y EL RESIDUO DEBEN PE-
    SAR MAS QUE EL CRISOL VACIO" : FOR K = 1 TO 1000 : NEXT
    K : GOTO 450
```

Mediante estas instrucciones se introducen los resultados de las medidas y se rechazan los incorrectos.

```
500 CLS
510 PRINT "RESULTADOS"
520 PRINT "1 – MASA DEL CRISOL ="; A; "G"
530 PRINT "2 – MASA DEL CRISOL + SOLIDO ="; B; "G"
540 S1 = B – A : S = INT (S1 * 100)/100
550 PRINT "3 – MASA DEL SOLIDO ="; B; "–"; A; "="; S; "G"
560 FOR K = 1 TO 1000 : NEXT K
570 CLS
580 F1 = C – A : F = INT (F1 * 100)/100
590 T1 = S – F : T = INT (T1 * 100)/100
600 PRINT "MASA DEL RESIDUO ="; F; "G"
610 PRINT "PERDIDA DE MASA ="; T; "G"
620 W1 = (T/S) * 100 : W = INT (W1 * 100)/100
630 PRINT "% PERDIDA DE MASA"; T; "/"; S; " * 100"; "="; W
640 FOR K = 1 TO 1000 : NEXT K
```

Este bloque de instrucciones realiza las operaciones necesarias, redondeando a dos cifras e imprime en la pantalla los resultados.

```
700  CLS
710  PRINT "RESULTADOS FINALES" : PRINT : PRINT
720  PRINT "MASA DEL CRISOL ="; A; "G"
730  PRINT : PRINT "MASA DEL SOLIDO ="; S; "G"
740  PRINT : PRINT "MASA DEL RESIDUO ="; F; "G"
750  PRINT : PRINT "PERDIDA DE MASA POR CALENTAMIENTO =";
     T; "G"
760  PRINT "PORCENTAJE DE PERDIDA DE MASA ="; W; "%"
770  END
```

Estas últimas instrucciones imprimen los resultados de las medidas tomadas en el experimento y calculan el porcentaje de pérdida de masa.

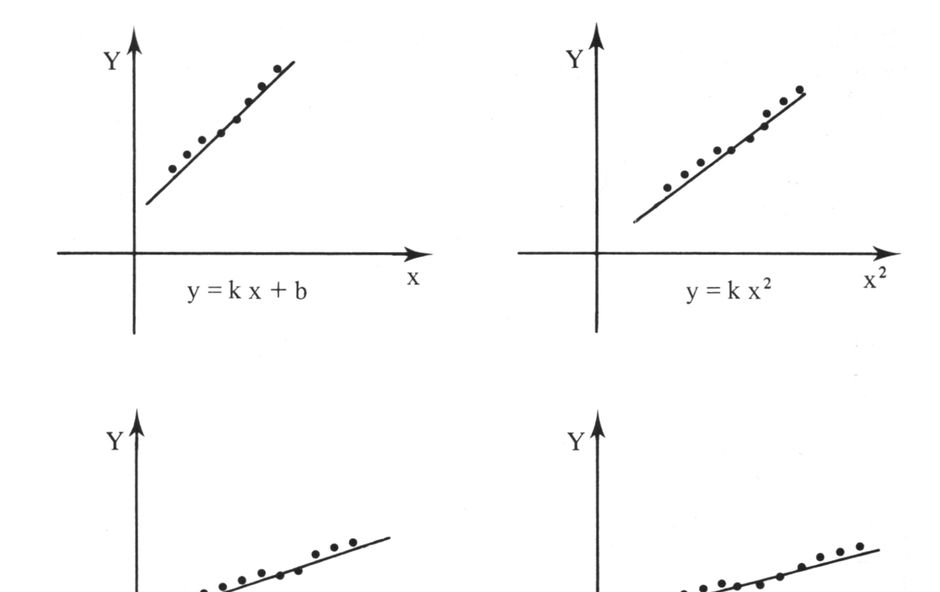
2. Análisis gráfico de datos experimentales

Presentación del problema

Antes de realizar un determinado experimento en el laboratorio, se establece la hipótesis de la existencia de una relación entre dos variables X, Y. Estas variables pueden ser, por ejemplo, la ***presión*** y el ***volumen*** de una masa de un gas, la ***velocidad*** y el ***tiempo*** en la caída de un cuerpo, la ***diferencia de potencial*** y la ***intensidad*** en un circuito, etc.

Para encontrar la relación entre las variables X, Y se registra una serie de datos en el curso del experimento, correspondientes a ambas variables, y después se dibuja la gráfica, Y – X. Esta permite observar si entre ambas variables existe o no relación y, en caso afirmativo, determinar la expresión matemática de la misma. Por ejemplo, si los puntos de la gráfica pertenecen a una línea recta, la relación entre las variables X, Y es una proporcionalidad, cuya expresión matemática es **Y = K X + + B,** siendo K la constante de proporcionalidad y B el término independiente.

Cuando la gráfica Y – X no es una línea recta, se tantean otras posibilidades, dibujando por ejemplo las gráficas $Y - X^2$, $Y - 1/X$, $Y - 1/X^2$, ... Si la gráfica obtenida es una recta, la relación entre X, Y será $Y = K X^2$, $Y = K (1/K)$, $Y = K (1/X^2)$, ..., respectivamente, siendo K la constante de proporcionalidad.



El programa

El programa que se transcribe permite encontrar la relación, si existe, entre las variables X, Y, analizando distintas gráficas.

```
 10  PRINT "ANALISIS DE DATOS"
 20  PRINT : PRINT
 30  INPUT "CUANTOS DATOS"; N
 40  DIM X (N) : DIM Y (N) : DIM Z (N) : DIM H (N) : DIM V (N)
 50  FOR I = 1 TO N
 60  INPUT "X ="; X (I) : INPUT "Y ="; Y (I)
 70  NEXT I
 80  CLS
 90  PRINT "X", "Y"
100  PRINT
110  FOR I = 1 TO N
120  PRINT X (I), Y (I)
130  NEXT I
140  PRINT "PULSA UNA TECLA"
150  A$ = INKEY$
160  IF A$ = "" THEN GOTO 150
```

Con estas instrucciones se introducen los datos (medidas del experimento), y se presentan a continuación en la pantalla en forma de tabla.

```
170  CLS
180  PRINT "PUEDES ELEGIR ENTRE LAS GRAFICAS:"
190  PRINT : PRINT "1) Y - X"
200  PRINT : PRINT "2) Y - X ∧ 2"
210  PRINT : PRINT "3) Y - 1/X"
220  PRINT : PRINT "4) Y - 1/X ∧ 2"
230  INPUT A
240  IF A < 1 OR A > 4 THEN GOTO 170
250  ON A GOTO 1300, 1400, 1500, 1600
```

Con las instrucciones **190, 200, 210** y **220** se elige la opción que se desee y con las instrucciones **1300, 1400, 1500** y **1600** se obtienen los valores de la variable independiente que se van a representar (según la opción, la variable independiente, Z (X), será X, X^2, 1/X ó $1/X^2$).

```
300  REM DIBUJA
310  CLS
320  Z2 = Z (1) : Z1 = Z (1) : Y2 = Y (1) : Y1 = Y (1)
330  FOR I = 2 TO N
340  IF Z2 < Z (I) THEN Z2 = Z (I)
350  IF Z1 > Z (I) THEN Z1 = Z (I)
360  IF Y2 < Y (I) THEN Y2 = Y (I)
370  IF Y1 > Y (I) THEN Y1 = Y (I)
380  NEXT I
390  IF Z1 > 0 THEN Z1 = 0
400  IF Y1 > 0 THEN Y1 = 0
410  DZ = Z2 - Z1 : DY = Y2 - Y1
420  ZS = 240/DZ : YS = 175/DY
```

Estas instrucciones escogen el mayor y menor valor de las variables y con ellos se establecen las escalas que se van a utilizar.

La instrucción

```
430  OPEN "GRP:" FOR OUTPUT A$ # 1
```

abre el canal 1, permite imprimir textos en la pantalla de alta resolución.

```
440 SCREEN 2
450 FOR I = 1 TO N
460 H (I) = (Z (I) + ABS (Y1)) * ZS : V (I) = (Y (I) + ABS (Y1)) * YS
470 NEXT I
480 PSET (15 + H (I), 175 - V (1))
490 FOR I = 2 TO N
500 LINE (15 + H (I - 1), 175 - V (I - 1)) - (15 + H (I), 175 - V (I))
510 NEXT I
```

Con estas instrucciones se dibujan los puntos obtenidos y se unen entre sí con tramos rectos.

```
520 PSET (15, 175)
530 PRINT # 1, "PULSA E PARA ELEGIR"
535 PSET (15, 185)
537 PRINT # 1, "PULSA F PARA ACABAR"
540 A$ = INKEY$
550 IF A$ = "E" OR A$ = "e" THEN CLOSE # 1: SCREEN 0 : GOTO 170
560 IF A$ = "F" OR A$ = "f" THEN SCREEN 0 : END
570 GOTO 540
```

La instrucción **520** coloca el cursor en el punto (15, 175) de la pantalla de alta resolución, a partir del cual la instrucción **530** imprime el texto.

Las instrucciones **550** y **560** analizan la opción elegida.

```
1300 FOR I = 1 TO N
1310 Z (I) = X (I)
1320 NEXT I
1330 GOTO 300
1400 FOR I = 1 TO N
1410 Z (I) = X(I)∧2
1420 NEXT I
1430 GOTO 300
1500 FOR I = 1 TO N
1510 IF X (I) = 0 THEN PRINT "NO ES POSIBLE" : GOTO 170
1520 Z (I) = 1/X (I)
1530 NEXT I
1540 GOTO 300
1600 FOR I = 1 TO N
1610 IF X (I) = 0 THEN PRINT "NO ES POSIBLE" : GOTO 170
1620 Z (I) = 1/X(I)∧2
1630 NEXT I
1640 GOTO 300
```

Estos distintos bloques de instrucciones calculan los valores de la variable independiente Z (I) en función de los datos X (I) introducidos al principio del programa.

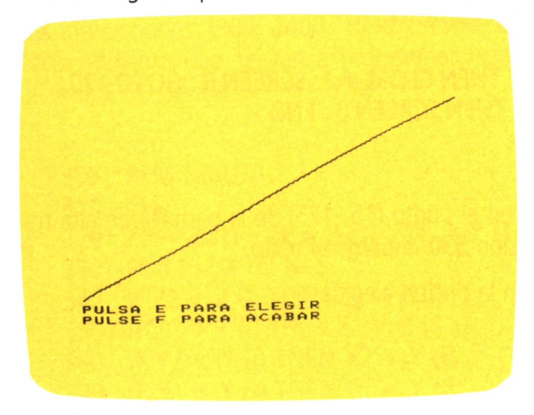
Dos ejecuciones del programa

- Supongamos que las medidas registradas en un experimento son las indicadas en la tabla.

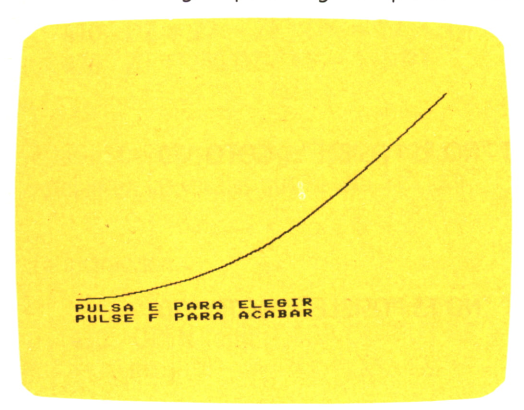
X	0	5	10	15	20	25	30	35	40	45	50
Y	0	6	29	63	110	170	252	345	450	565	702

Después de teclear **RUN** y RETURN se procede a introducir estos datos.

Si se elige la opción 1 se obtiene el resultado que muestra la fotografía.



Como esta gráfica no se aproxima a una línea recta, en lugar de pulsar F para acabar, se elige E para elegir la opción 2. Se obtiene este otro resultado:



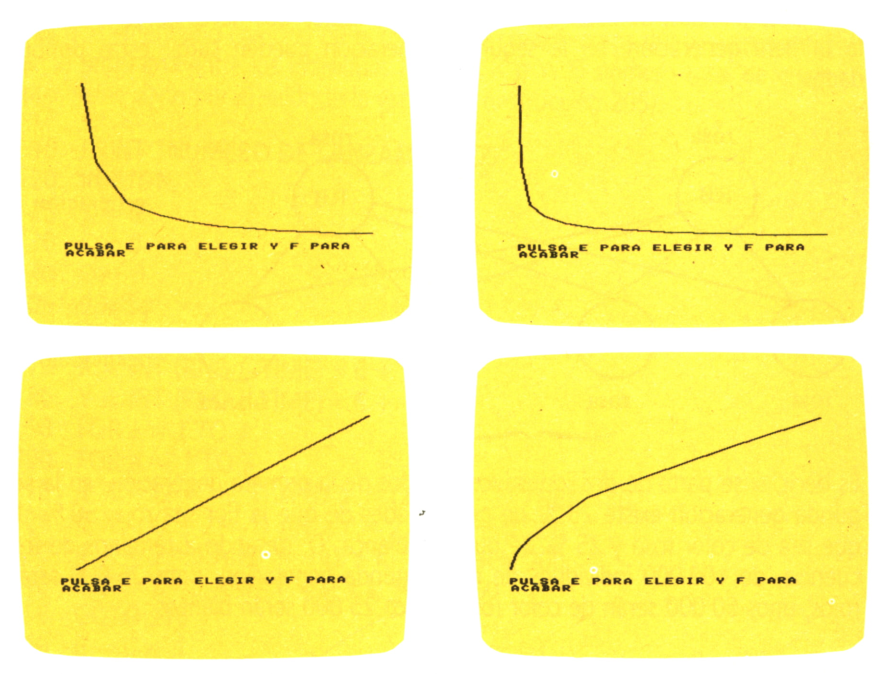
Esta gráfica se aproxima mucho a una recta. Se deduce, entonces, que la relación es del tipo **$Y = K * X^2$**, siendo K la constante de proporcionalidad (*).

- Supongamos, ahora que las medidas registradas en otro experimento son éstas:

X	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
Y	11.00	5.50	3.85	2.80	2.25	1.90	1.55	1.40	1.25	1.14

Después de teclear **RUN** y **ENTER** se procede a introducir los datos registrados en la tabla.

Si se ejecutan las cuatro opciones (pulsar sucesivamente las teclas 1, 2, 3 y 4) se obtienen estas gráficas.



Observando estas gráficas, se llega a la conclusión que la relación entre X, Y es del tipo **$Y = K (1/X)$**, siendo **K** la constante de proporcionalidad (**).

(*) Esta constante se puede determinar dividiendo un valor de Y por el cuadrado de su correspondiente valor de X: $Y = K * X^2 \Rightarrow K = Y/X^2$

(**) Esta constante se puede determinar multiplicando un valor de Y por su correspondiente valor de X: $Y = K * (1/X) \Rightarrow Y * X = K$

12. Simulación de fenómenos aleatorios en biología y física

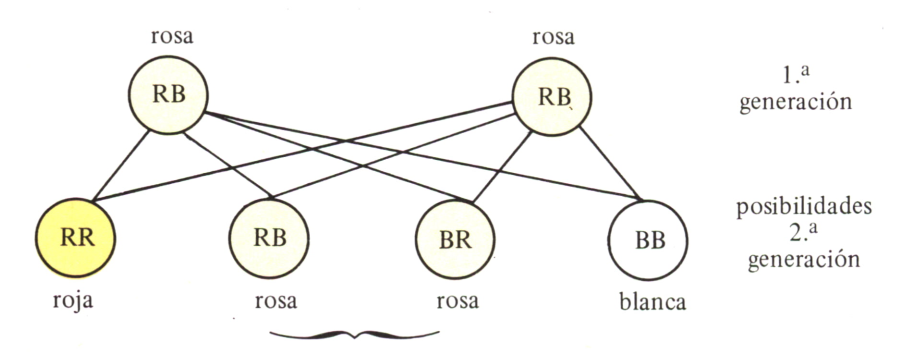
1. Estudio de leyes genéticas

Presentación del problema

Vamos a analizar una de las leyes genéticas: la ***segunda ley de Mendel.***

Recordaremos esta ley mediante un ejemplo.

Si se cruzan flores rojas (RR) con flores blancas (BB), en la primera generación surgen flores híbridas, de color rosa (RB) (*). Al cruzarse dos flores de color rosa de la primera generación, en la segunda generación pueden surgir estas posibilidades:

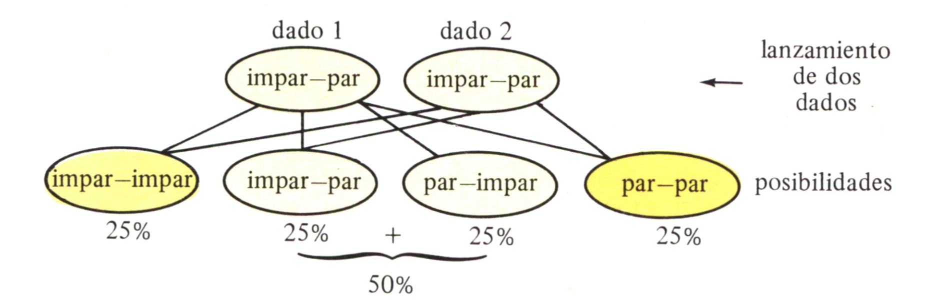


Es decir, si se parte de dos individuos híbridos de la primera generación, en la segunda generación existe 25 % de posibilidades de que la flor sea roja, 50 % de que sea de color rosa y 25 % de que sea blanca. O, pasando a términos de frecuencia, de 100 000 individuos de esta segunda generación, unos 25 000 serán rojos, unos 50 000 serán de color rosa y unos 25 000 serán blancos.

La comprobación experimental de esta ley lleva mucho tiempo y exige mucho trabajo. Pero con ayuda del microordenador se pueden simular los experimentos y comprobar rápidamente la ley. Para ello simularemos el lanzamiento de dos datos X, Y. Cada uno de ellos representa una flor híbrida, en cuanto que contiene, por decirlo así, dos genes diferentes: impar (R) y par (B).

(*) RR y BB representan genotipos de dos razas puras, cada uno de ellos con dos genes R, R y B, B. El genotipo RB es híbrido, por tener dos genes diferentes R y B.

Al lanzar los dos dados pueden presentarse cuatro posibilidades:



El programa

El número N de «lanzamientos» a que hacen referencia las instrucciones **10** y **20** no hay que entenderlo como el número total de cruzamientos que se van a simular en el programa entre los individuos híbridos X e Y. El número total de cruzamientos viene dado por el valor de la variable T (instrucción **205**).

```
10  PRINT "NUMERO DE LANZAMIENTOS"
20  INPUT N
25  T = 0
30  C = 0
40  D = 0
50  E = 0
60  FOR I = 1 TO N
70  X = INT (RND (-TIME) * 6 + 1)
80  Y = INT (RND (-TIME) * 6 + 1)
90  FOR J = 1 TO 3
100 FOR K = 1 TO 3
110 IF X = (2 * J - 1) AND Y = (2 * K - 1) THEN 180
120 IF X = (2 * J - 1) AND Y = 2 * K THEN 200
130 IF X = 2 * J AND Y = (2 * K - 1) THEN 200
140 IF X = 2 * J AND Y = 2 * K THEN 160
150 GOTO 210
160 C = C + 1
170 GOTO 210
180 D = D + 1
190 GOTO 210
200 E = E + 1
205 T = T + 1
210 NEXT K
220 NEXT J
230 NEXT I
```

En los bucles anidados se simula el lanzamiento de un dado N veces y se obtienen los resultados X e Y de cada lanzamiento.

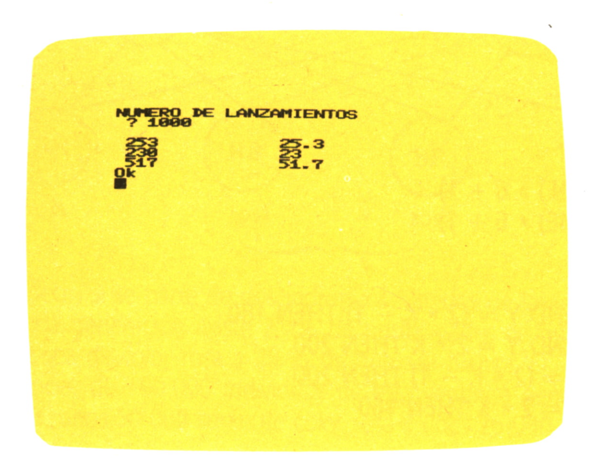
En los bucles correspondientes a las instrucciones **90** a **220** se cuenta el número de veces que sale cada combinación, teniendo en cuenta que la combinación ***pares - impares*** se repite.

```
240 PRINT C, C * 100/N
250 PRINT D, D * 100/N
260 PRINT E, E * 100/N
270 END
```

Finalmente, con estas instrucciones se imprimen los resultados C, D y E y los porcentajes que representan, respectivamente.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introduce 1000 para N se obtienen estos resultados.



2. Vida

Planteamiento del juego

Este es un juego muy conocido que simula el nacimiento, crecimiento y muerte de una colonia de células.

Las células iniciales se colocan al azar en una cuadrícula de 10 × 10 y nacen, viven y mueren según las siguientes reglas:

— Cada célula de la retícula tiene 7 vecinas.

— Cada célula con 2 ó 3 vecinas sobrevive a la siguiente generación.

— Si hay 3 y sólo 3 células próximas nace una nueva célula.

— Cualquier célula con 4 o más vecinas muere por haber exceso de población.

El programa

A continuación se simula lo que les ocurre a diferentes generaciones de células hasta llegar a una situación estable (una situación puede consistir en la muerte de todas las células).

```
10  CLS
20  G = 0
30  DIM A (10, 10), B (10, 10)
40  FOR X = 2 TO 9
50  FOR Y = 2 TO 9
60  A = RND (–TIME)
70  IF A > 0.35 THEN A (X, Y) = 1
80  B (X, Y) = A (X, Y)
90  NEXT Y
100 NEXT X
105 PRINT
```

Con estas instrucciones se plantea una situación inicial.

```
110 GOSUB 1000
120 G = G + 1
130 FOR X = 2 TO 9
140 FOR Y = 2 TO 9
145 C = 0
150 IF A (X – 1, Y – 1) = 1 THEN C = C + 1
160 IF A (X – 1, Y) = 1 THEN C = C + 1
170 IF A (X – 1, Y + 1) = 1 THEN C = C + 1
180 IF A (X, Y – 1) = 1 THEN C = C + 1
190 IF A (X, Y + 1) = 1 THEN C = C + 1
200 IF A (X + 1, Y – 1) = 1 THEN C = C + 1
210 IF A (X + 1, Y) = 1 THEN C = C + 1
220 IF A (X + 1, Y + 1) = 1 THEN C = C + 1
```

La instrucción **110** transfiere el control a una subrutina que dibuja la situación inicial, la instrucción **120** establece el paso de las generaciones y las siguientes, analizan la situación de cada célula.

```
230  IF A (X, Y) = 1 AND C <> 3 AND C <> 2 THEN B (X, Y) = 0
240  IF A (X, Y) = 0 AND C = 3 THEN B (X, Y) = 1
250  NEXT Y
260  NEXT X
280  GOTO 105
```

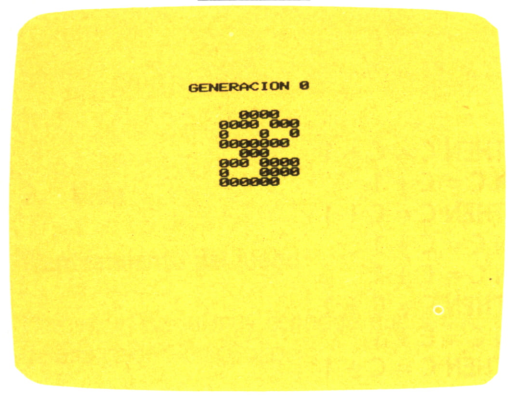
La instrucción **230** produce la muerte de una célula, mientras que la **240** produce su nacimiento.

```
1000  LOCATE 9, 3 : PRINT "GENERACION"; G
1010  FOR X = 1 TO 10
1020  FOR Y = 1 TO 10
1030  A (X, Y) = B (X, Y)
1040  IF A (X, Y) = 1 THEN LOCATE X + 10, Y + 4 : PRINT "0"
1050  IF A (X, Y) = 0 THEN LOCATE X + 10, Y + 4 : PRINT " "
1060  NEXT Y
1070  NEXT X
1080  RETURN
```

Esta subrutina construye la figura adecuada a las distintas situaciones.

Ejecución del problema

Tecleando **RUN** y RETURN se obtiene este resultado.



3. Simulación de la desintegración radiactiva

Planteamiento del problema

Supongamos una determinada cantidad de monedas que lanzamos al aire. Una vez que éstas han caído, quitamos las que presentan «cruz» anotando las «caras». Lanzamos al aire las monedas restantes (las «caras») retirando las «cruces» y anotando de nuevo las que han salido «cara». Repetimos el procedimiento varias veces.

La variación en cada lanzamiento del número de caras puede utilizarse como modelo para analizar la variación en el tiempo del número de átomos que no se desintegran en un proceso radiactivo.

El programa

El programa que se incluye a continuación hace esta simulación con la ayuda de un microordenador. Permite obtener el número de átomos que no se han desintegrado todavía en cada instante, y pulsando **CONT** se obtiene la representación gráfica de este número en función del tiempo.

```
 2 CLS
 5 DIM N (500), H (500), V (500), D (500)
10 INPUT "NUMERO INICIAL DE MONEDAS"; N
20 D (0) = 0 : J = 0 : N (0) = N
30 FOR I = 1 TO N (J)
40 S = INT (RND (–TIME) + 0.5)
50 IF S = 0 THEN GOTO 70
60 D (J) = D (J) + 1
70 NEXT I
80 N (J + 1) = N (J) – D (J)
90 IF N (J) > = 2 THEN J = J + 1 : GOTO 30
```

Estas instrucciones simulan el lanzamiento de N monedas, asignando el valor 1 a las caras y el 0 a las cruces. En cada lanzamiento se contabilizan las caras.

A continuación se plantea un nuevo lanzamiento retirando las caras que se habían obtenido anteriormente (instrucción **80**).

```
100 FOR U = 0 TO J
110 PRINT U, N (U)
120 NEXT U
130 STOP
```

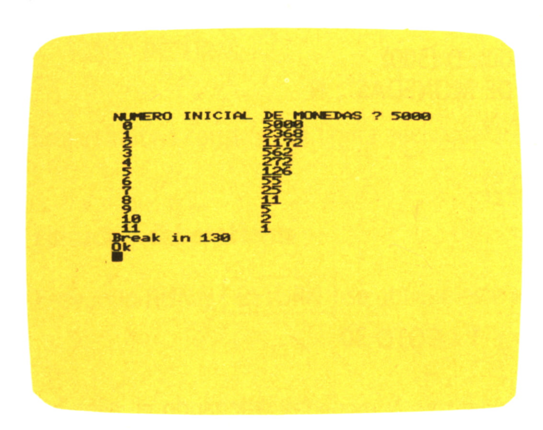
La instrucción **110** imprime en la pantalla el número de átomos sin desintegrar (o monedas que presentan cruz), en función del tiempo.

```
140 DX = J - 0 : DY = N (0) - N (J)
150 XS = 245/DX : YS = 180/DY
160 SCREEN 2
170 FOR U = 0 TO J
180 H (U) = U * XS : V (U) = (N (U) + N (J)) * YS
190 NEXT U
200 PSET (H (0), 175 - V (0))
210 FOR U = 1 TO J
220 LINE (J (U - 1), 175 - V (U - 1)) - (H (U), 175 - V (U))
230 NEXT U
240 GOTO 240
```

Una vez que se pulsa **CONT** se obtiene la representación gráfica del número de átomos sin desintegrar en función del tiempo.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introduce 5000 para N se obtiene esta pantalla.



4. Equilibrio térmico de un sólido

Presentación del problema

El equilibrio térmico de un sólido puede simularse con el microordenador mediante el siguiente juego:

Se construye un tablero de 800 casillas, por ejemplo, y en cada una de ellas se coloca una ficha.

Se lanzan dos dados hipotéticos de 800 caras. El resultado del lanzamiento del primer dado señalará de qué casilla se retira una ficha, mientras que el del segundo, señalará el lugar en el que debe colocarse dicha ficha.

Si la casilla obtenida en el primer lanzamiento no contiene ficha, se repite el lanzamiento, pero contabilizándolo entre los realizados.

Si se hace un número suficiente de lanzamientos la distribución del número de casillas que contienen 0, 1, 2, 3, 8 fichas es exponencial.

Los resultados concuerdan en cierta manera con los que se obtienen en la distribución cuántica de un sólido en equilibrio térmico.

El programa

A continuación se simula este juego.

```
 5 KEY OFF
10 CLS
20 DIM C (8), H (8), V (8)
30 INPUT "CUANTOS MOVIMIENTOS"; A
40 X = 50000
50 FOR I = 0 TO 800
60 POKE X + I, 49
70 NEXT I
```

Con la instrucción **POKE** se coloca un cuanto en las 800 posiciones de memoria siguientes a la 50 000 (el código ASCII del número 1 es precisamente 49).

```
80 GOSUB 2000
```

Mediante esta subrutina se simula el lanzamiento de un dado.

```
 90 Z = PEEK (X + D)
100 IF Z < 49 THEN GOTO 80
110 POKE X + D, Z − 1
120 GOSUB 2000
130 B = PEEK (X + D)
140 POKE X + D, B + 1
150 J = J + 1
160 IF J = A THEN GOTO 1000
170 GOTO 80
```

En la instrucción **90** se coge una ficha de una casilla elegida al azar (subrutina **2000**). Si la casilla no contiene ficha se repite el lanzamiento (instrucción **80**).

Después de simular de nuevo el lanzamiento del dado (instrucción **120**), de acuerdo con el resultado que salga se añade una ficha a la posición obtenida (instrucción **140**).

El proceso se repite hasta alcanzar el valor de A (número de lanzamientos).

```
1000  FOR T = 0 TO 800
1010  FOR I = 0 TO 8
1020  IF PEEK (X + T) = 48 + I THEN C (I) = C (I) + 1
1030  NEXT I
1040  NEXT T
```

Con este bloque de instrucciones se contabilizan las casillas que contienen 0 fichas (48 es el código ASCII de 0), 1 ficha (49 es el código ASCII de 1), y así sucesivamente hasta 9 fichas (48 + 8 es el código ASCII de 8).

```
1100  FOR I = 0 TO 8
1110  PRINT "EL NUMERO DE ATOMOS CON"; I; "CUANTOS ES"; C (I)
1120  NEXT I
1130  STOP
```

Estas instrucciones imprimen en la pantalla el número de átomos con 0, 1, 2, ..., 8 cuantos.

```
1200  Y2 = C (0) : Y1 = C (0)
1210  FOR I = 1 TO 8
1220  IF Y2 < C (I) THEN Y2 = C (I)
1230  IF Y1 > C (I) THEN Y1 = C (I)
1240  NEXT I
1250  IF Y1 > 0 THEN Y1 = 0
1260  DX = 8 : DY = Y2 - Y1
1270  XS = 245/DX : YS = 175/DY
1280  SCREEN 2
1290  FOR I = 0 TO 8
1300  H (I) = I * XS : V (I) = (C (I) + ABS (Y1)) * YS
1310  NEXT I
1320  PSET (H (0), 175 - V (0))
1330  FOR I = 1 TO 8
1340  LINE (H (I - 1), 175 - V (I - 1)) - (H (I), 175 - V (I))
1350  NEXT I
1360  GOTO 1360
```

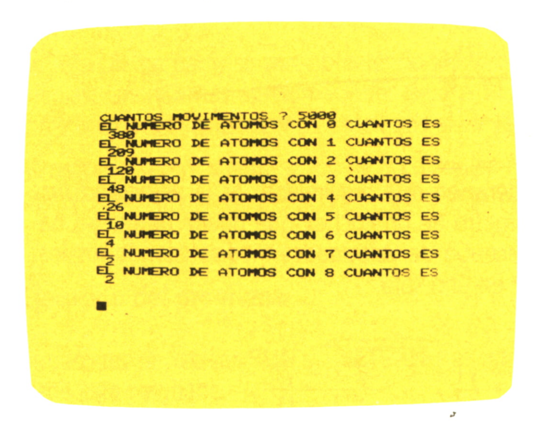
Utilizando un procedimiento que se ha usado varias veces a lo largo de este libro, se representa el número de casillas en función del número de cuantos que hay en dichas casillas. La gráfica aparece después de pulsar la tecla **CONT**.

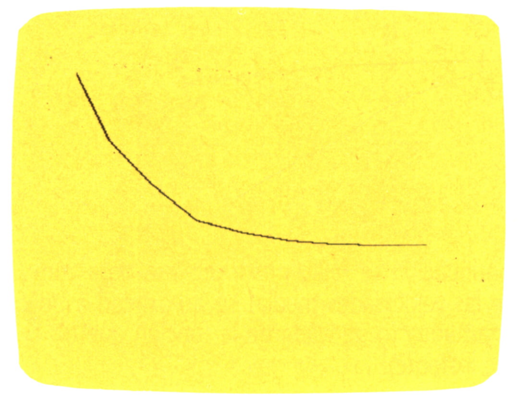
```
2000  D = INT (800 * RND (–TIME))
2010  RETURN
```

Ésta es la subrutina que elige al azar la casilla.

Ejecución del programa

A pesar de utilizar instrucciones **POKE** y **PEEK,** la ejecución del programa es bastante lenta, debido al gran número de casillas que elige y al número de movimientos que realiza. Así, para 5 000 movimientos, la ejecución puede durar aproximadamente unos diez minutos. Si se introduce, entonces, 5 000 para A, después del tiempo indicado se obtiene este resultado.





13. Cálculo numérico en Física y Biología

1. Análisis del concepto de velocidad instantánea (derivada de una función en un punto)

Presentación del programa

Consideremos un móvil que se mueve siguiendo una trayectoria cualquiera. Su ***velocidad media*** entre dos instantes t1 y t2, en los que ocupa dos posiciones y1 e y2 (con respecto a un punto de referencia 0), se obtiene dividiendo el espacio recorrido por el intervalo de tiempo que tarda en recorrerlo:

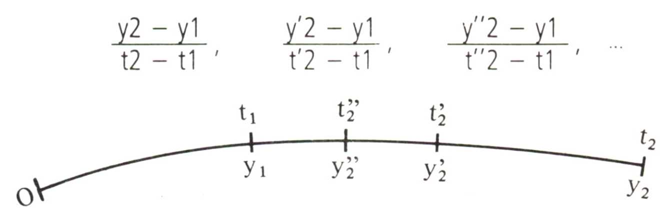
$$vm = (y2 - y1) / (t2 - t1)$$

O y_1 t_1 y_2 t_2

Otra velocidad que se puede considerar en este movimiento es la ***velocidad instantánea*** (la que marcaría el velocímetro de un coche), que puede obtenerse hallando el límite al que tienden la sucesión de velocidades medias cuando los instantes t1 y t2 tienden a confundirse en uno solo.

La sucesión se expresa así:

$$\frac{y2 - y1}{t2 - t1}, \quad \frac{y'2 - y1}{t'2 - t1}, \quad \frac{y''2 - y1}{t''2 - t1}, \quad \ldots$$



El programa

En el programa que se incluye a continuación se realiza este proceso para comprobar finalmente que los valores de las velocidades medias se aproximan a uno determinado, que es la velocidad instantánea (o derivada de la función que da el espacio con respecto al tiempo en ese instante).

El procedimiento permite calcular, con aproximación suficiente, la derivada de cualquier función en un punto.

```
 1 REM VELOCIDAD INSTANTANEA
10 CLS
20 DEF FN S (T) = T ↑ 2 + 2 * T + 3
30 INPUT "INTRODUCE EL INSTANTE"; T1
```

En estas instrucciones se define la función que relaciona el espacio con el tiempo y se introduce el instante en el que se va a obtener la velocidad.

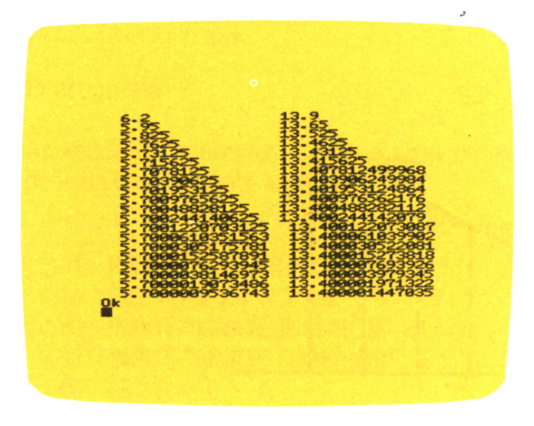
```
35 CLS
40 FOR N = 1 TO 20
50 T = T1 + 0.5 ↑ N
60 V = (FN S (T) – FN S (T1)) / T – T1
70 PRINT TAB (1); T; TAB (17); V
80 NEXT N
90 END
```

En esta última parte se van obteniendo los términos de la sucesión de velocidades medias. Las distintas aproximaciones de T a T1 se obtienen sumando a T1 los valores cada vez más pequeños obtenidos al elevar 0.5 a los diversos valores de N.

El procedimiento puede aplicarse a cualquier función matemática que se desee, por ejemplo, a $y = e^{t}$, $y = e^{-t}$, y = 35 en t y a otras más complicadas.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introduce el valor 5.7 para T1, se obtiene este resultado.



2. Cálculo del espacio recorrido en un movimiento (Cálculo numérico de integrales definidas)

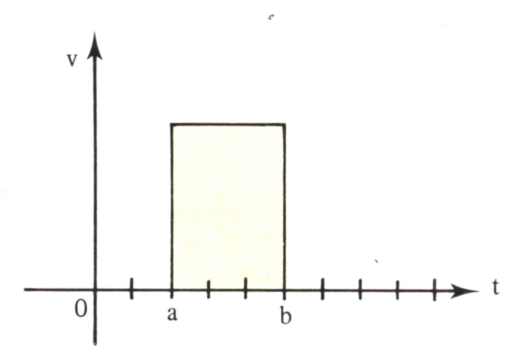
Presentación del problema

En un movimiento rectilíneo y uniforme, el espacio recorrido entre dos instantes a y b viene expresado por la fórmula

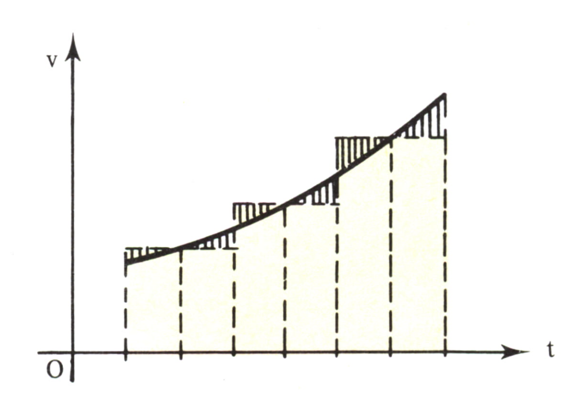
$$S = V\,(b - a)$$

donde S es el espacio recorrido, V es la velocidad (constante) y b − a, el intervalo de tiempo transcurrido.

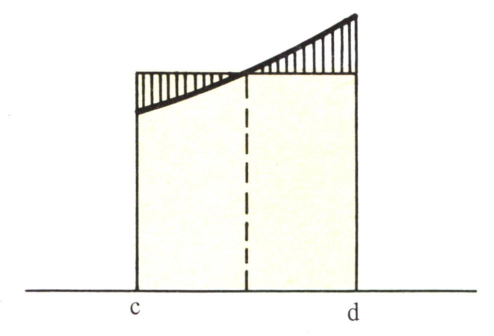
En un diagrama velocidad − tiempo el espacio recorrido corresponderá al área del rectángulo de altura v y base b − a.



Si el movimiento no es uniforme, la gráfica ***velocidad − tiempo*** no es una línea recta paralela al eje t, con lo que el cálculo del espacio recorrido se hace más complejo. Para hacer este cálculo se divide el intervalo a − b en intervalos parciales c − d, y en cada uno de estos intervalos se aproxima el movimiento real a un movimiento uniforme con velocidad constante e igual a la del móvil real en la mitad del intervalo.

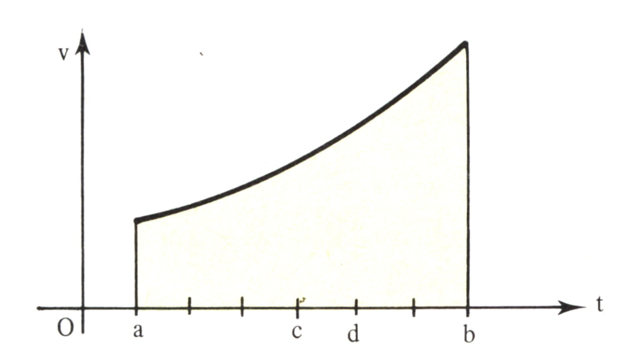


El espacio recorrido en cada intervalo c – d, en el movimiento ficticio, es igual al área del rectángulo destacado en color.



La suma de los espacios recorridos en todos los intervalos c – d se aproximará tanto más al espacio total recorrido en el movimiento real cuanto menor sea la amplitud de los intervalos c – d, lo cual equivale a que estén más próximos entre sí los tramos escalonados.

Formando una sucesión de sumas, así calculadas, para intervalos de tiempo cada vez más pequeños, dicha sucesión tiende a un valor límite: el espacio recorrido por el móvil en el movimiento, o sea la ***integral definida de la velocidad en el intervalo de tiempo a – b.***



El programa

Aplicando los conceptos que se acaban de exponer, se logra calcular la integral definida con suficiente aproximación.

```
 1 REM ESPACIO RECORRIDO
10 DEF FN V (T) = 2 * T ↑ 2 + 3
20 INPUT "LIMITE INFERIOR"; A
30 INPUT "LIMITE SUPERIOR"; B
40 LET D = B – A
```

En esta parte del programa se define la función velocidad y se introducen los tiempos entre los cuales se va a calcular el espacio recorrido, A y B.

Se define también una variable D, que se inicia con el valor B − A y que en el curso de la ejecución del programa se irá dividiendo por 2 para obtener su mitad.

```
 50  CLS
 60  FOR I = 1 TO 10
 70  LET M = 0
 80  FOR T = A + D/2 TO B STEP D
 90  LET M = M + FN V (T)
100  NEXT T
110  LET M = M * D
120  PRINT M
130  LET D = D/2
140  NEXT I
150  END
```

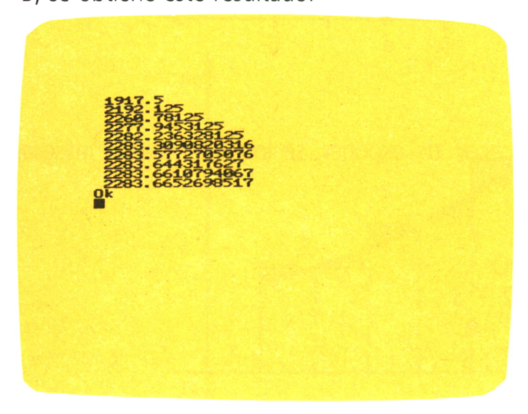
Este bloque de instrucciones calcula diez términos de la sucesión de sumas, las cuales se van acumulando en la variable M. Cada suma se obtiene en el bucle interno (instrucciones **80** a **100**) y en la instrucción **110.**

Para cada valor de I que determina el bucle externo (**60** a **140**), se divide D por la mitad.

El último valor de M que se imprime en la pantalla es el más aproximado al espacio recorrido por el móvil real.

Ejecución del programa

Tecleando **RUN** y RETURN e introduciendo el valor 2 para A y el valor 15 para B, se obtiene este resultado.



3. Cálculo del número de células nacidas en un determinado tiempo

Presentación del problema

El número de células de una población puede duplicarse si dichas células se reproducen dividiéndose en dos, y siempre que el alimento sea el adecuado.

La relación $\frac{dN}{dt} = \mathbf{N}$, que expresa que la variación del número de células por unidad de tiempo es igual al número de células existentes en un instante (número inicial de células), representa bien este fenómeno.

Escribiendo $\frac{dN}{dt} = \mathbf{N}$ en forma de incrementos finitos se obtiene:

$$\frac{\Delta N}{\Delta t} = N \quad \text{o bien} \quad \frac{N\,(t + \Delta t) - N(t)}{\Delta t} = N$$

Despejando N (t + Δt) se consigue esta ecuación:

$$\mathbf{N\,(t + \Delta t) = N\,(t) \times \Delta t + N\,(t)}$$

Si, por ejemplo, N (0) = 1 y Δt = 1, se tiene:

$$N\,(1) = N\,(0) * 1 + N\,(0) = 1 + 1 = 2$$
$$N\,(2) = N\,(1) * 1 + N\,(1) = 2 + 2 = 4$$
$$N\,(3) = N\,(2) * 1 + N\,(2) = 4 + 4 = 8$$

Se ve cómo, dando valores a t, el número de células se duplica en cada unidad de tiempo. Así:

- al término de la primera unidad de tiempo el número de células es **2**;
- al término de la segunda unidad de tiempo el número de células es **4**;
- al término de la tercera unidad de tiempo el número de células es **8**;
- etcétera.

El programa

El programa que se transcribe a continuación imprime el número de células que hay en cada instante y después de teclear **CONT** dibuja la gráfica ***número de células – tiempo.***

```
10  DIM T (100), N (100), H (100), V (100)
20  CLS
30  INPUT "NUMERO INICIAL DE CELULAS"; NO
40  INPUT "TIEMPO"; TM
```

En estas instrucciones se introducen los datos necesarios para que el programa pueda ejecutarse.

```
50  I = 0 : N (0) = NO : IT = 1 : T (0) = 0
60  N (I + 1) = N (I) * IT + N (I)
70  T (I + 1) = T (I) + IT
80  IF T (I + 1) < = TM THEN I = I + 1 : GOTO 60
```

Este grupo de instrucciones determina el número de células existentes en un instante a partir de las existentes en el instante anterior. En la instrucción **70** se incrementa el tiempo en IT y se comprueba si se ha alcanzado el tiempo previsto.

```
 90  FOR J = 0 TO I + 1
100  PRINT T (J), N (J)
110  NEXT J
120  STOP
```

Mediante estas instrucciones se imprimen en la pantalla los tiempos y el número de células.

En la línea **120** se detiene la ejecución. Para que la ejecución continúe hay que teclear **CONT** RETURN .

```
130  X2 = T (0) : X1 = T (0) : Y2 = N (0) : Y1 = N (0)
140  FOR J = 1 TO I + 1
150  IF X2 < T (J) THEN X2 = T (J)
160  IF X1 > T (J) THEN X1 = T (J)
170  IF Y2 < N (J) THEN Y2 = N (J)
180  IF Y1 > N (J) THEN Y1 = N (J)
190  NEXT J
200  IF X1 > 0 THEN X1 = 0
210  IF Y1 > 0 THEN Y1 = 0
220  DX = X2 - X1 : DY = Y2 - Y1
230  XS = 245/DX : YS = 175/DY
```

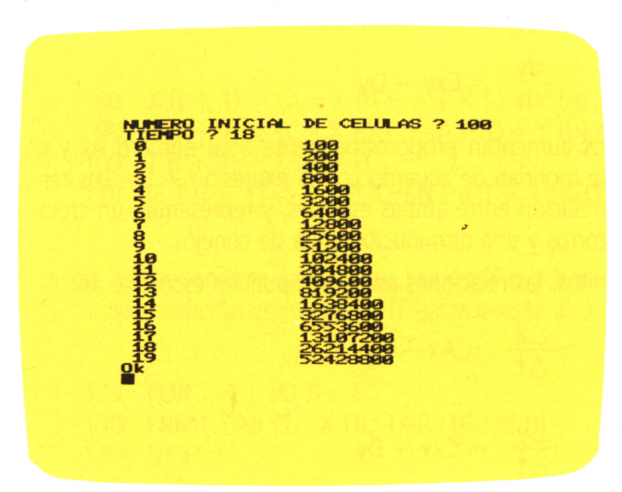
Cuando se escribe **CONT** se buscan los valores mínimos y máximo de T ( ) y N ( ), y se establecen las escalas adecuadas para dibujar la gráfica N – T.

```
240 SCREEN 2
250 FOR J = 0 TO I + 1
260 H (J) = (T (J) + ABS (T (0))) * XS : V (J) = (N (J) + ABS (N (0))) *
    YS
270 NEXT J
280 PSET (H (0), 175 – V (0))
290 FOR J = 1 TO I + 1
300 LINE (H (J – 1), 175 – V (J – 1)) – (H (J), 175 – V (J))
305 NEXT J
310 GOTO 310
```

Este bloque de instrucciones calcula y dibuja los puntos, y a continuación los une mediante tramos rectos.

Ejecución del programa

Si después de teclear **RUN** y RETURN se introduce 100 para N y el valor 18 para T, se obtiene en la pantalla este resultado.



En la línea **120** la ejecución del programa está detenida. Tecleando **CONT** y RETURN sigue la ejecución. Al final de la misma se obtiene en la pantalla la gráfica N – T.

Observación

El programa que se acaba de ejecutar se puede utilizar, con ligeras modificaciones, para estudiar otros procesos, como la desintegración radiactiva, la descarga de un condensador, etc.; es decir, aquellos en que dos magnitudes F y X están relacionadas por una ecuación del tipo

$$\frac{dF}{dx} = KF$$

siendo K una constante.

4. Un problema de ecología

Presentación del problema

Estudiaremos ahora la evolución de dos especies de animales que coexisten en un mismo hábitat, alimentándose una especie de la otra.

Un ejemplo de estas características lo constituye una población de conejos y otra de zorros, donde los segundos se alimentan de los primeros.

Si x es el número de conejos e y el número de zorros, se admite que el tamaño de las dos poblaciones evoluciona según estas leyes:

$$\frac{dx}{dt} = Ax - Bxy$$

$$\frac{dy}{dt} = Cxy - Dy$$

Si no hay zorros, los conejos aumentan proporcionalmente a su número Ax y si no hay conejos, los zorros se morirían de acuerdo con la expresión $- Dy$. Los términos en xy reflejan la interrelación entre ambas especies, y representan un crecimiento en la población de zorros y una disminución en la de conejos.

En forma de incrementos finitos, las relaciones anteriores pueden escribirse así:

$$\frac{\Delta x}{\Delta t} = Ax - Bxy$$

$$\frac{\Delta y}{\Delta t} = Cxy - Dy$$

Despejando $x (t + \Delta t)$ e $y (t + \Delta t)$, se obtiene:

$$x (t + \Delta t) = (Ax (t) - Bx (t) \cdot y (t)) \Delta t + x (t)$$
$$y (t + \Delta t) = (Cx (t) y (t) - Dy (t)) \Delta t + y (t)$$

El programa

Mediante estas expresiones se puede averiguar la población X de conejos y la población Y de zorros en cada instante.

Esto es lo que calcula el siguiente programa, que además dibuja en una misma pantalla (después de teclear **CONT** y RETURN) las gráficas X − T e Y − T, lo que permite comparar la evolución de ambas especies.

En este programa se eligen como valores adecuados los siguientes valores para las constantes:

A = 1, B = 2, C = 2 y D = 3.

```
 1  REM ECOLOGIA
10  DIM T (500), X (500), Y (500), H (500), V (500), Z (500)
20  A = 4 : B = 2 : C = 1 : D = 3
30  CLS
40  INPUT "NUMERO INICIAL DE CONEJOS"; XO
50  INPUT "NUMERO INICIAL DE ZORROS"; YO
60  INPUT "TIEMPO"; MT
70  IT = 0.01 : X (0) = XO : Y (0) = YO : T (0) = 0 : I = 0
```

Aquí se plantean las condiciones iniciales y se introducen los datos necesarios para que el programa pueda ejecutarse.

```
 80  X (I + 1) = (A * X (I) − B * X (I) * Y (I)) * IT + X (I)
 90  Y (I + 1) = (C * X (I) * Y (I) − D * Y (I)) * IT + Y (I)
100  T (I + 1) = T (I) + IT
110  IF T (I + 1) < = MT THEN I = I + 1 : GOTO 80
```

Estas instrucciones se calculan los valores X e Y a partir de sus valores anteriores. El tiempo aumenta en intervalos IT hasta que se alcance el valor MT.

```
120  FOR J = 1 TO I + 1
130  PRINT TAB (1); X (J); TAB (16); Y (J)
140  NEXT J
145  STOP
```

El bucle **120-130** imprime en la pantalla los valores sucesivos de X e Y, y la instrucción **145** detiene la ejecución del programa.

```
150  X2 = X (0) : Y2 = Y (0) : T2 = T (0) : X1 = X (0) : Y1 = Y (0) :
     T1 = T (0)
160  FOR J = 1 TO I+1
170  IF T2 < T (J) THEN T2 = T (J)
180  IF T1 > T (J) THEN T1 = T (J)
190  IF X2 < X (J) THEN X2 = X (J)
200  IF X1 > X (J) THEN X1 = X (J)
210  IF Y2 < Y (J) THEN Y2 = Y (J)
220  IF Y1 > Y (J) THEN Y1 = Y (J)
230  NEXT J
240  IF X1 > 0 THEN X1 = 0
250  IF Y1 > 0 THEN Y1 = 0
260  IF T1 > 0 THEN T1 = 0
270  DT = T2 - T1 : DX = X2 - X1 : DY = Y2 - Y1
280  TS = 245/DT : XS = 180/DX : YS = 245/DY
```

Después de escribir **CONT** se buscan los valores máximo y mínimo de las distintas variables y se establecen las escalas para dibujar.

```
290  SCREEN 2
300  FOR J = 0 TO I + 1
310  H (J) = (T (J) + ABS (T (0))) * TS : V (J) = (Y (J) + ABS (Y (0))) *
     YS : Z (J) = (X (J) + ABS (X (0))) * XS
320  NEXT J
330  PSET (H (0), 175 - V (0))
340  FOR J = 1 TO I + 1
350  LINE (H (J - 1), 175 - V (J - 1)) - (H (J), 175 - V (J))
360  NEXT J
370  PSET (H (0), 175 - Z (0))
380  FOR J = 1 TO I + 1
390  LINE (H (J - 1), 175 - Z (J - 1)) - (H (J), 175 - Z (J))
400  NEXT J
410  GOTO 410
```

Este conjunto de instrucciones dibuja en una misma pantalla las gráficas ***número de conejos – tiempo*** y ***número de zorros – tiempo.***

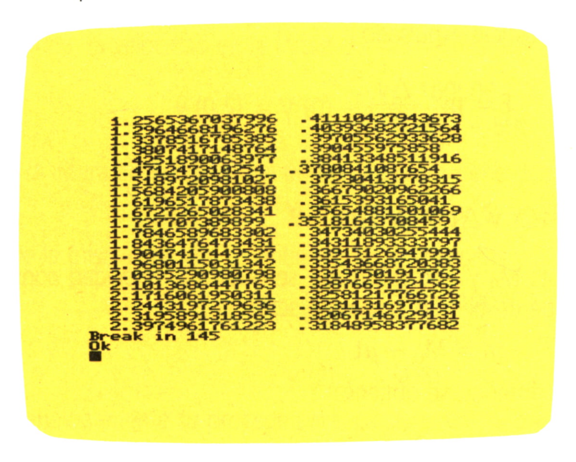
Los valores iniciales de conejos XO y los valores iniciales de zorros YO, así como el tiempo MT, deben ser números pequeños para que no se produzca error (OVER FLOW).

Ejecución del programa

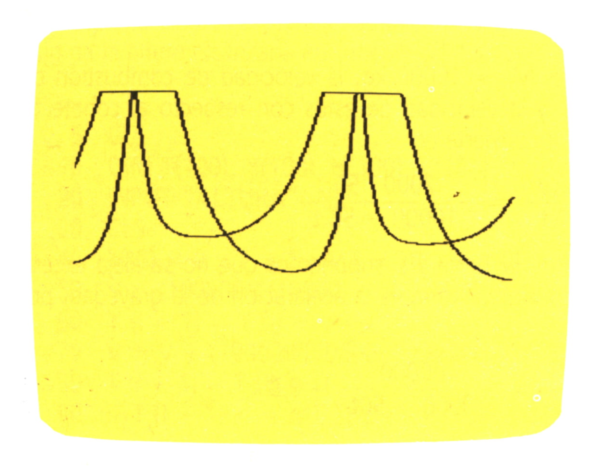
Después de teclear **RUN** y RETURN introducir los valores que se indican a continuación para las correspondientes variables:

XO = 3, YO = 0.5, MT = 4.2 (*)

En la pantalla se obtiene este resultado:



En la línea **145** se detiene la ejecución del programa. Para que continúe hay que teclear **CONT** RETURN, obteniéndose las gráficas que se observan en esta pantalla.



(*) Los valores 3 y 0.5 no indican que se parta inicialmente de 3 conejos y de 0.5 zorros, sino que la relación inicial del número de conejos a zorros es 3 a 0.5. El tiempo MT = 4.2 tampoco es el real, sino el que corresponde a una cierta escala.

5. Análisis del movimiento de un cohete de masa variable

Consideremos un cohete cuya masa en un instante es m y que, quemando una masa $\triangle m$ durante un tiempo $\triangle t$, aumenta su velocidad en $\triangle v$. Supongamos también que los gases procedentes de esta combustión tienen una velocidad u con respecto al cohete.

La fuerza media que actúa sobre el cohete debe ser igual a la variación de la cantidad de movimiento del combustible expulsado.

$$F = m \frac{\triangle v}{\triangle t}$$

Es decir,

$$m \triangle v/\triangle t = -v \triangle m/\triangle t$$

Si la masa inicial del cohete es M_0 y el combustible se quema a velocidad constante $\mu = \triangle m/\triangle t$, después de un tiempo t, la masa será:

$$m = M_0 - \mu t$$

y sustituyendo en la ecuación anterior, se obtiene:

$$(M_0 - \mu t) \frac{\triangle v}{\triangle t} = u\mu$$

O bien

$$\frac{\triangle v}{\triangle t} = \frac{u\mu}{M_0 - \mu t}$$

Si la masa inicial del cohete es $M_0 = 10000$ kg, la velocidad de combustión de los mismos es $\mu = 50 \text{ kgs}^{-1}$, y la velocidad de éstos con respecto al cohete es $u = 2000 \text{ ms}^{-1}$, la expresión se convierte en

$$\frac{\triangle v}{\triangle t} = \frac{2000 \cdot 50}{10000 - 50\,t}$$

Si el cohete está despegando de la Tierra y si suponemos que no se aleja excesivamente de ésta como para considerar variable la aceleración de la gravedad, podemos escribir:

$$\frac{\triangle v}{\triangle t} = \frac{100000}{10000 - 50t} - 9.8$$

Teniendo en cuenta que $\frac{\triangle v}{\triangle t}$ se puede escribir como

$$\frac{v\,(t + \triangle t) - v\,(t)}{\triangle t}$$

despejando, se obtiene:

$$v(t + \triangle t) = v(t) + \left(\frac{100000}{10000 - 50\,t} - 9.8\right) \triangle t$$

La altura viene dada por

$$y(t + \triangle t) = y(t) + v(t) \triangle t$$

Si la velocidad inicial es V (0) = 0, considerando intervalos de tiempo de un segundo, la velocidad en la mitad del primer intervalo es

$$v(0.5) = V(0) + \left(\frac{100000}{10000 - 50t} - 9.8\right) * 0.5$$

y la altura alcanzada en el primer segundo viene dada por

$$Y(1) = V(0.5) \times 1 + y(0)$$

De la misma manera se obtienen las expresiones v (1.5) e y (2):

$$v(1.5) = V(0.5) + \left(\frac{100000}{10000 - 50t} - 9.8\right)$$

$$y(2) = v(1.5) \times 1 + y(1)$$

y análogamente se procede para los sucesivos valores del tiempo.

El programa

El programa siguiente permite obtener una representación gráfica de la función que da la altura alcanzada en función del tiempo.

```
 5  CLS
10  DIM T(100), Y(100), H(100), V(100)
20  INPUT "ALTURA"; YO
30  CLS
40  T (0) = 0 : IT = 1 : MT = 20 : V = 0 : I = 0
50  V = (100000 / (10000 - 50 * IT) - 9.8) * IT/2 + V
60  Y (I + 1) = Y (I) + V * IT
70  V = V + (100000/(10000 - 50 * T) - 9.8) * IT + V
80  T (I + 1) = T (I) + IT
90  IF T (I + 1) < = MT THEN I = I + 1 : GOTO 40
```

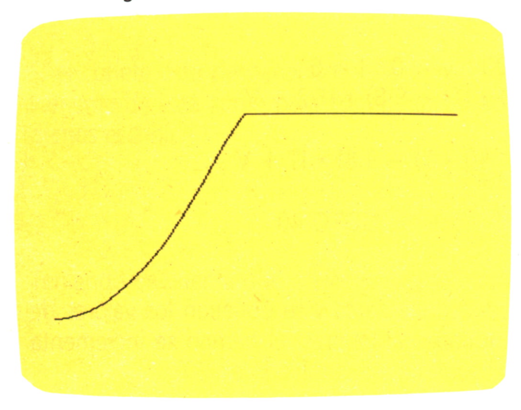
Mediante estas instrucciones se introducen los datos y se obtienen los valores de la altura a partir de las alturas y velocidades anteriores. El tiempo se incrementa en intervalos I hasta llegar al límite MT (que aquí se ha fijado en 20).

```
100  X2 = T (0) : X1 = T (0) : Y2 = Y (0) : Y1 = Y (0)
110  FOR J = 1 TO I + 1
120  IF X2 < T (J) THEN X2 = T (J)
130  IF X1 > T (J) THEN X1 = T (J)
140  IF Y2 < Y (J) THEN Y2 = Y (J)
150  IF Y1 > Y (J) THEN Y1 = Y (J)
160  NEXT J
170  IF X1 > 0 THEN X1 = 0
180  IF Y1 > 0 THEN Y1 = 0
190  DX = X2 − X1 : DY = Y2 − Y1
200  XS = 245/DX : YS = 175/DY
205  SCREEN 2
210  FOR J = 0 TO I + 1
220  H (J) = (T (J) + ABS (T (0))) * XS : V (J) = (Y (J) + ABS (Y (0))) *
     XS
230  NEXT J
240  PSET (H (0), 175 − V (0))
250  FOR J = 1 TO I + 1
260  LINE (H (J − 1), 175 − V (J − 1)) − (H (J), 175 − V (J))
270  NEXT J
280  GOTO 280
```

Con estas instrucciones se seleccionan los valores máximo y mínimo de las variables, se establece la escala y se realiza el dibujo uniendo los puntos con tramos rectos.

Ejecución del programa

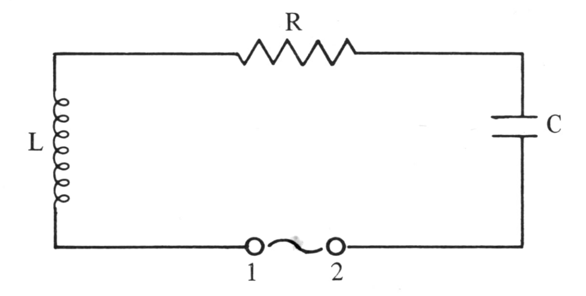
Si después de teclear **RUN** y RETURN se introduce el valor 100 para YO, se obtiene esta gráfica.



6. Estudio de las oscilaciones en un circuito eléctrico de corriente alterna

Presentación del problema

Consideremos un circuito con un condensador C, una resistencia R y una autoinducción L, dispuestos en serie.



El voltaje V (t) aplicado en los bornes 1 y 2 es igual al trabajo necesario para trasladar la unidad de carga desde el punto 1 al 2, a través de todos los elementos del circuito. Este trabajo se obtiene sumando los siguientes valores:

— La caída de potencial en los extremos de la bobina:

$$V_L = L \frac{dI}{dt} = L \frac{d^2 Q}{dt}$$

— La caída de potencia en los extremos de la resistencia:

$$V_R = R I = R\, dQ / dt$$

— La caída de potencial en los extremos del condensador:

$$V_C = Q/C$$

Luego V (t) viene dado por

$$V(t) = L\, d^2 Q/d t^2 + R\, d Q/d t + Q/C$$

Si el voltaje varía sinusoidalmente en función del tiempo, se tiene

$$V(t) = V_0 \text{ sen } (2 \pi F t)$$

donde V_0 es el voltaje máximo, F la frecuencia y t el tiempo transcurrido.

Reemplazando V (t) por V_0 sen (2 π F t) en la ecuación anterior se obtiene esta expresión:

$$L \frac{dI}{dt} + RI + \frac{Q}{C} = V_0 \text{ sen } (2 \pi F T)$$

Despejando $\frac{dI}{dt}$, se tiene:

$$\frac{dI}{dt} = V_0/L \text{ sen } (2 \pi F t) - RI/L - Q/LC$$

Y considerando incrementos finitos se tendrá:

$$\triangle I/\triangle t = V_0/L \text{ sen } (2 \pi F t) - RI/L - Q/LC$$

O bien, teniendo en cuenta que

$$\frac{\triangle I}{\triangle t} = \frac{I(t + t) - I(t)}{\triangle t}$$

se obtiene:

$$I(t + \triangle t) = I(t) + (V_0/L \text{ sen } (2 \pi F t) - RI/L - Q/LC) \triangle t$$

Pero como I = △ Q/ △ t, se llega finalmente a la expresión:

$$Q(t + \triangle t) = Q(t) + I \cdot \triangle t$$

Teniendo en cuenta las expresiones anteriores se calcula la carga para intervalos de tiempo sucesivos y la intensidad en la mitad de dichos intervalos. Así, por ejemplo, para los intervalos 0 – 0.1 y 0.1 – 0.2, se obtiene:

$$I(0.05) = I(0) + V_0/L * \text{sen } (2 \pi FT \times 0.1) - (RI(0)/L) \cdot (0.1) - Q(0)/LC) \times (0.1)$$

y

$$Q(0.1) = Q(0) + I(0.05) \times (0.1)$$

$$I(0.15)=I(0.05)+V_0/L \text{ sen } (2 \pi F t\times 0.1)-(RI(0.05)/L)\times 0.1-Q(0.05)/LC) \times 0.1$$

y

$$Q(0.2) = I(0.15) \times (0.1) + Q(0.1)$$

El programa

El programa tiene en cuenta los conceptos que se acaban de exponer y dibuja la gráfica *intensidad – tiempo.*

```
5  CLS
10 INPUT "RESISTENCIA"; R
20 INPUT "INDUCTANCIA"; L
30 INPUT "CAPACIDAD"; C
40 INPUT "VOLTAJE EFICAZ"; V
50 INPUT "FRECUENCIA"; F
60 W = 2 * 3.14 * F
65 P = 1/F
70 XC = 1/(C * W)
80 XL = L * W
85 Q = 0
90 Z = SQR (R ↑ 2 + (XL – XC) ↑ 2)
100 VO = V * SQR (2)
110 IO = VO/R
```

Con estas instrucciones se determinan, a partir de los datos introducidos, los valores de la impedancia Z, del voltaje máximo VO en función del voltaje eficaz ($v_{máx.} = V_{ef.} \cdot \sqrt{2}$) y de la intensidad máxima.

```
120 T1 = 0 : T2 = 3 * P
125 Y1 = –IO : Y2 = IO
130 DT = (T2 – T1)/245 : DY = (Y2 – Y1)/180
140 SCREEN 2
150 I = VO * SIN (2 * 3.14 * F * T) * DT / (2 * L) – R * I* DT / (2 * L) –
    – Q * DT / (2 * L * C) + I
160 X3 = INT ((T – T1) / DT + 0.5)
170 Y3 = INT ((I – Y1) / DY + 0.5)
175 LINE (0,90) – (250,90)
176 LINE (15,0) – (15,180)
180 PSET (15 + X3, 180 – Y3)
190 Q = Q + Q * DT
200 I = VO * SIN (2 * 3.14 * F * T) * DT / L – R * I * DT / L – Q *
    DT / (L * C) + I
210 T = T + DT
220 IF T < = T2 THEN GOTO 160
250 GOTO 250
```

En esta parte del programa se van trazando los distintos puntos, obtenidos por el procedimiento recursivo que ha sido descrito en los comentarios que anteceden al programa. Las instrucciones **175** y **176** trazan los ejes de coordenadas.

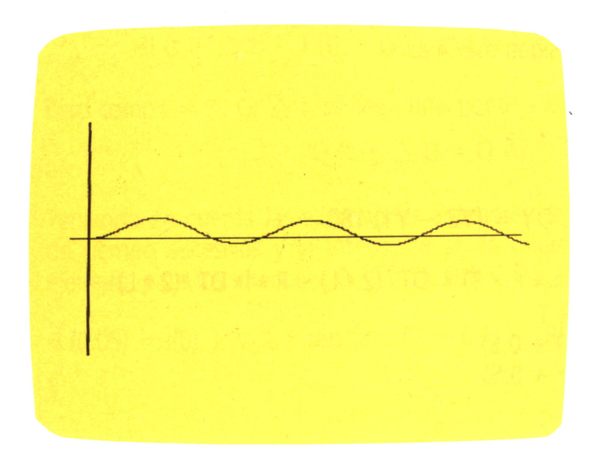
Ejecución del programa

Proponemos se ejecute el programa para dos situaciones: la primera con valores cualesquiera de R, L, C, V y F, y la segunda, introduciendo el valor de F que corresponda a la resonancia del circuito, es decir, para $F = \frac{1}{2} \sqrt{L C}$.

- ***Primera ejecución***

Después de teclear **RUN** y RETURN si se introducen los valores:

300 para R
98 para L
0.5 para C
100 para V
5 para F
se obtiene esta gráfica:



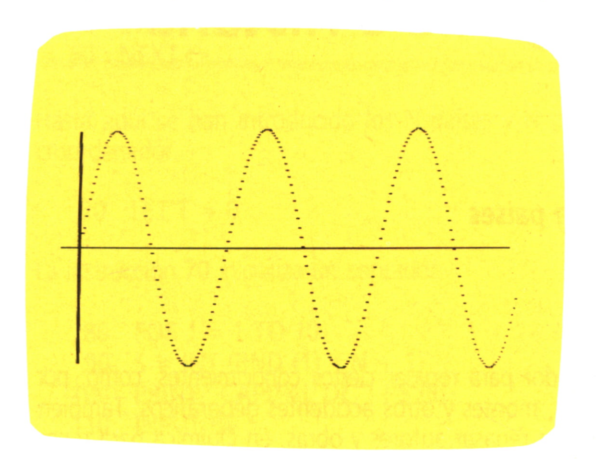
- ***Segunda ejecución***

En esta ejecución introducimos estos valores:

300 para R
98 para L
0.5 para C
100 para V
y para F, la frecuencia de resonancia:

$$F = \frac{1}{2*3.14*\sqrt{98*0.5}} = \frac{1}{2*3.14*7} = 0.023$$

La gráfica que se obtiene es la que muestra la pantalla.



14. Geografía e Historia

1. Repaso de capitales de países

Presentación del problema

Se puede utilizar el microordenador para repasar ciertos conocimientos, como, por ejemplo, capitales de países, ríos, montes y otros accidentes geográficos. También puede utilizarse en Literatura para repasar autores y obras, en Química para repasar nombres y símbolos de elementos químicos, etc.

El programa

En primer lugar se introducen en la memoria, como variables alfanuméricas con índices, los países y capitales: P$ (I), C$ (I).

Para dar cada contestación se introduce otra variable alfanumérica, B$. El número N de países que se introducen varía a voluntad.

De todos los países y capitales se extraerá una al azar escogiendo su índice (S).

La respuesta se comparará con el contenido de la variable C$ (I). En caso de error se dará el resultado correcto, y en caso de acierto se sumará una unidad a un contador.

```
 1  REM AVERIGUA CAPITALES
10  Z = RND (–TIME)
15  INPUT N
20  DIM P$ (N)
25  DIM C$ (N)
```

La instrucción **10** hace que la instrucción **90** genere números aleatorios. Si el microordenador no es MSX la instrucción **10** se deberá suprimir o, en todo caso, habrá que sustituirla por otra, según el tipo de microordenador que se esté utilizando.

La instrucción **15** pide el número de países y capitales, y las instrucciones **20** y **25** dimensionan las variables de cadena P$ y C$.

```
30 FOR I = 1 TO N
40 INPUT P$ (I)
50 INPUT C$ (I)
60 NEXT I
```

Hasta aquí se han introducido los N países y N capitales en la memoria del microordenador.

```
70 LET T = 0
```

La instrucción **70** inicializa un contador.

```
80 FOR J = 1 TO 10
90 S = INT (RND (1) * N + 1)
100 PRINT "CAPITAL DE"; P$ (S)
110 INPUT B$
120 PRINT B$
130 IF B$ = C$ (S) THEN GOTO 160
140 PRINT "TE EQUIVOCASTE; EL RESULTADO ES:"; C$ (S)
150 GOTO 180
160 T = T + 1
170 PRINT "ACERTASTE"
180 NEXT J
```

En esta parte se obtiene un índice S al azar y se compara la capital correspondiente con la respuesta dada. Con el bucle **80-180** se repite el proceso 10 veces.

```
190 PRINT "EL NUMERO DE ACIERTOS ES"; T
200 END
```

Se imprime finalmente el número de aciertos (contenido del contador T).

2. Pirámides de población

Presentación del problema

En Geografía se plantea frecuentemente el problema de la distribución de la población de un país por edades y sexo. La representación gráfica de esta distribución es lo que llamamos *pirámide de población.*

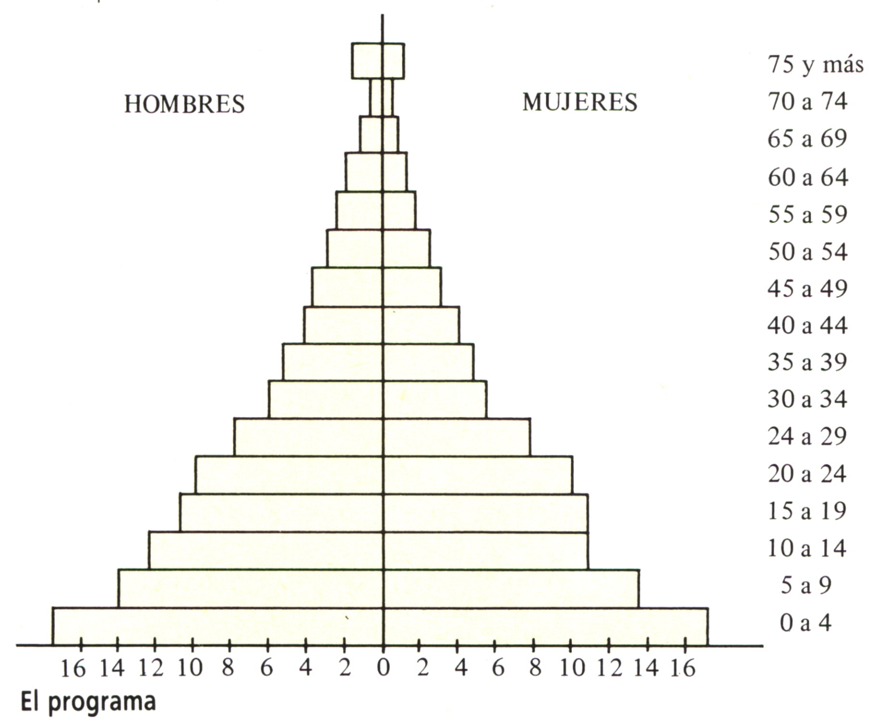
Así, por ejemplo, la población de Guatemala en el año 1960 era de 3.965.000 habitantes aproximadamente, y su distribución por edades, de cinco en cinco, era la siguiente:

Grupo de edades (años)	*Número de hombres (miles)*	*Número de mujeres (miles)*
0 a 4	321,6	344,3
5 a 9	262,6	278,3
10 a 14	232,2	233,0
15 a 19	201,7	233,0
20 a 24	182,7	212,4
25 a 29	148,4	165,0
30 a 34	114,2	117,5
35 a 39	106,2	115,5
40 a 44	76,1	90,7
45 a 49	73,3	76,3
50 a 54	56,1	63,9
55 a 59	41,9	41,2
60 a 64	38,1	39,2
65 a 69	19,0	18,6
70 a 74	11,4	14,4
75 y más	17,1	18,6

El total de hombres era igual a 1.903.000 y el total de mujeres 2.062.000. A partir de estos datos se calculan las frecuencias relativas correspondientes, dividiendo el número de hombres (o mujeres) de un grupo determinado de edades entre el total de hombres (o mujeres). La tabla de porcentajes que se obtiene es ésta:

Grupos de edades (años)	*Frecuencia relativa (hombres)*	*Frecuencia relativa (mujeres)*
0 a 4	0,169	0,167
5 a 0	0,138	0,135
10 a 15	0,122	0,113
15 a 19	0,106	0,113
20 a 24	0,096	0,103
25 a 29	0,078	0,080
30 a 34	0,060	0,057
35 a 39	0,056	0,056
40 a 44	0,040	0,044
45 a 49	0,038	0,037
50 a 54	0,030	0,031
55 a 59	0,022	0,020
60 a 64	0,020	0,019
65 a 69	0,010	0,009
70 a 74	0,006	0,007
75 y más	0,009	0,009

Con estos datos se puede construir la pirámide de población correspondiente, colocando en la parte izquierda los nombres y en la derecha las mujeres. Al eje vertical se ha llevado la edad de cinco en cinco años y al eje horizontal la frecuencia relativa porcentual (frecuencia relativa × 100).



El programa

Los elementos necesarios para diseñar un programa que dé la pirámide de población son los siguientes:

1.° Dos matrices H y M, de dimensión 16 cada una (una dimensión por cada grupo de edad), para almacenar los datos de los grupos de edad de hombres y mujeres, respectivamente. Además, dos totalizadores de hombres y mujeres TH y TM.

2.° Dos bucles para ir asignando los valores de los distintos grupos de edad: primero, a la matriz H y, segundo, a la matriz M. En estos bucles se contalizarán todos los hombres, TH, y todas las mujeres, TM.

3.° Un bucle en el que se calculen los porcentajes (H/TH y M/TM); como estos nuevos valores van a estar entre 0 y 0.20 aproximadamente, los multiplicamos por 50 para que se encuentren entre 0 y 10, pues de esta forma es más fácil representarlos gráficamente.

H(K) = (H(K)/TH) * 50
M(K) = (M(K)/TM) * 50

Por último, en este mismo bucle, estos valores se redondean al entero más próximo. Es decir, si, por ejemplo, H(3) = 5.6 tomamos H(3) = 6 y si H(4) = = 3.5 tomamos H(4) = 3.

4.° El paso siguiente consiste en dibujar los ejes, y en la parte inferior izquierda poner HOMBRES y en la parte inferior derecha MUJERES.

5.° Por medio de dos bucles anidados se dibujan en la parte izquierda de la gráfica los puntos correspondientes a los hombres y, en la parte derecha, los correspondientes a las mujeres.

```
 1 REM PIRAMIDE DE POBLACION
10 DIM H(16)
20 DIM M(16)
30 TH = 0
40 TM = 0
50 I = 0
60 J = 4
```

En las instrucciones **10** y **20** se dimensionan còn el valor 16 las matrices de grupos de hombres y de grupos de mujeres H y M. En las **30** y en la **40** se ponen a cero los contadores *total hombres* TH y *total mujeres* TM. Por último, en la **50** se pone a cero el contador izquierdo de los grupos de edad y la **60** inicia en cuatro el contador derecho (0 a 4 años).

```
 70 FOR K = 1 TO 16
 80 PRINT "HOMBRES DE"; I; "A"; J; "AÑOS"
 90 INPUT H(K)
100 TH = TH + H(K)
110 I = I + 5
120 J = J + 5
130 NEXT K
140 CLS
```

La instrucción **80** imprime el grupo de edad que hay que introducir, introducción que se hace en la instrucción **90.** En la **100** el total de hombres TH se incrementa con el valor introducido en **90.** Las instrucciones **110** y **120** incrementan los límites del grupo de edad en cinco. La **130** repite el bucle hasta que se introduzcan los 16 valores. La **140** borra la pantalla.

```
150  I = 0
160  J = 4
170  FOR K = 1 TO 16
180  PRINT "MUJERES DE"; I; "A"; J; "AÑOS"
190  INPUT M(K)
200  TM = TM + M(K)
210  I = I + 5
220  J = J + 5
230  NEXT K
240  CLS
```

Las instrucciones **150** a **160** vuelven a iniciar los límites de grupos de edad I y J, para los valores de mujeres. Desde la instrucción **170** a la **230** se repite el bucle anterior, sólo que para mujeres. Se termina borrando la pantalla con la instrucción **240.**

```
250  FOR K = 1 TO 16
260  H(K) = (H(K)/TH) * 50
270  IF H(K) - INT (H(K)) > 0.5 THEN H(K) = H(K) + 1
280  H(K) = INT (H(K))
290  M(K) = (M(K)/TM) * 50
300  IF M(K) - INT (M(K)) > 0.5 THEN M(K) = M(K) + 1
310  M(K) = INT (M(K))
320  NEXT K
```

En la instrucción **250** comienza el bucle que termina en la **320.** En la **260** se calcula el porcentaje de cada grupo de hombres respecto del total de hombres; como este valor que se obtiene está entre 0 y 1, se multiplica por 50 para luego poder hacer la representación gráfica. En la **270** se observa si la parte decimal de H(K) es mayor que 0.5; en caso afirmativo, se suma una unidad. La instrucción **280** deja a H(K) con su parte entera, eliminando la parte decimal. En las instrucciones **290, 300** y **310** se realiza el mismo proceso para los grupos de mujeres.

```
330  LOCATE 7,21 : PRINT "HOMBRES"
340  LOCATE 17,21 : PRINT "MUJERES"
350  FOR K = 0 TO 31
360  LOCATE K, 19 : PRINT "█"
370  NEXT K
380  FOR K = 0 TO 18
390  LOCATE 15,K : PRINT "█"
400  NEXT K
```

Las instrucciones **330** y **340** escriben las palabras HOMBRES y MUJERES en la parte inferior del gráfico. El bucle **350-370** dibuja el eje horizontal y el **380-400** el eje vertical.

```
410  FOR K = 1 TO 16
420  FOR L = 0 TO H(K)
430  LOCATE 15-L, 19-K : PRINT "█"
440  NEXT L
450  NEXT K
```

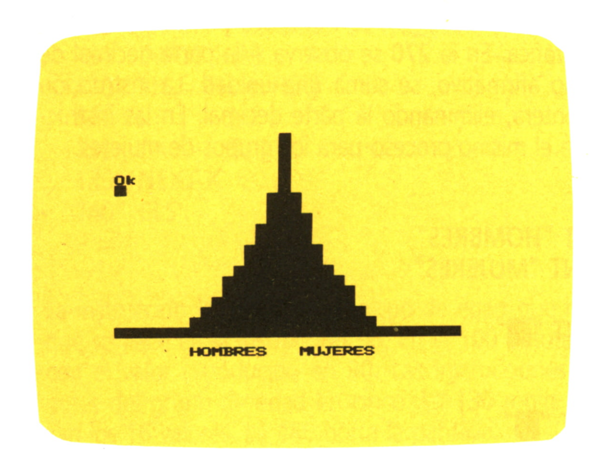
Estos bucles anidados dibujan la parte de la pirámide de población correspondiente a los 16 grupos de edades de hombres.

```
460  FOR K = 1 TO 16
470  FOR L = 0 TO M(K)
480  LOCATE 15 + L, 19-K : PRINT "█"
490  NEXT L
500  NEXT K
```

Análogamente al bloque anterior, éste dibuja la pirámide correspondiente a los 16 grupos de edades de mujeres.

Ejecución del programa

Si se ejecuta el programa introduciendo como datos los del ejemplo anterior (Guatemala - 1960) se obtiene el siguiente resultado en la pantalla:



Observación

Si se quieren mantener los datos junto con el programa, eliminar las instrucciones **80, 90, 180** y **190,** e introducir las siguientes:

```
 80  READ H(K)
180  READ M(K)
510  DATA 321.6, 262.6, 232.2, 201.7, 182.7, 148.4, 114.2, 106.2, 76.1,
          73.3, 56.1, 41.9, 38.1, 19.0, 11.4, 17.1
520  DATA 344.3, 278.3, 233, 233, 212.4, 165.0, 117.5, 115.5, 90.7, 76.3,
          63.9, 41.2, 39.2, 18.6, 14.4, 18.6
```

3. Ordenar cronológicamente

Presentación del problema

El siguiente programa sirve para comprobar el acierto en una ordenación cronológica de acontecimientos. Serviría también en Literatura para situar en el tiempo a ciertos autores, así como para comprobar la ordenación de elementos según energías de ionización, radios atómicos, etc.

El programa

Inicialmente se introducen estos acontecimientos ordenadamente en la memoria. El número de acontecimientos introducidos N puede determinarse a voluntad. El número de acontecimientos que salen en pantalla al azar es tres, distintos entre sí.

Para evitar escribir demasiado, se colocan las letras A, B y C delante de ellos y éstas son las respuestas que deben darse.

Las respuestas se comparan con el contenido de la memoria y, en caso de error, el programa da la respuesta correcta.

```
10  INPUT N
20  DIM A$ (N)
30  DIM C (3)
40  DIM D$ (3)
```

Se declaran variables que se van a utilizar y se introduce el número de acontecimientos.

```
50  FOR I = 1 TO N
60  INPUT A$ (I)
70  NEXT I
```

Con este bucle se introducen ordenadamente los distintos acontecimientos.

```
75  D$ (1) = "A"
74  D$ (2) = "B"
76  D$ (3) = "C"
```

Se almacenan los valores A, B y C en D$ (1), D$ (2), D$ (3), respectivamente. Más adelante se compararán con las contestaciones.

```
 90  S = INT (RND (1) * N + 1)
100  T = INT (RND (1) * N + 1)
105  IF T = S THEN GOTO 100
110  X = INT (RND (1) * N + 1)
115  IF X = S OR X = T THEN GOTO 110
```

Se eligen tres respuestas al azar que sean distintas.

```
120  PRINT "A"; A$ (S)
130  PRINT "B"; A$ (T)
140  PRINT "C"; A$ (X)
```

Se imprimen los tres acontecimientos elegidos con las letras A, B y C delante.

```
150  FOR J = 1 TO 3
160  INPUT B$ (J)
170  PRINT B$ (J)
180  IF B$ (J) = D$ (1) THEN GOTO 220
190  IF B$ (J) = D$ (2) THEN GOTO 240
200  IF B$ (J) = D$ (3) THEN GOTO 260
210  GOTO 550
220  B$ (J) = A$ (S)
230  GOTO 270
240  B$ (J) = A$ (T)
250  GOTO 270
260  B$ (J) = A$ (X)
270  NEXT J
```

Pide una contestación A, B o C y después se asigna la contestación A$ (S), A$ (T) o A$ (X) a B$ (1), B$ (2) y B$ (3), respectivamente.

```
280  C(1) = S
290  C(2) = T
300  C(3) = X
310  FOR M = 1 TO 2
320  FOR N = 1 TO 2
330  X = C(N)
340  Y = C(N + 1)
350  IF X < = Y THEN GOTO 380
360  C(N) = Y
370  C(N + 1) = X
380  NEXT N
390  NEXT M
```

Ordena S, T y X en forma creciente (ver programa de ordenación de números).

```
400  S = C (1)
410  T = C (2)
420  X = C (3)
```

Coloca en S, T y X el valor correctamente ordenado.

```
430  IF B$ (1) = A$ (S) THEN GOTO 450
440  GOTO 500
450  IF B$ (2) = A$ (T) THEN GOTO 470
460  GOTO 500
470  IF B$ (3) = A$ (X) THEN GOTO 490
480  GOTO 500
490  PRINT "ACERTASTE"
495  GOTO 90
500  PRINT "TE EQUIVOCASTE, EL RESULTADO ES:"
510  PRINT A$ (S)
520  PRINT A$ (T)
530  PRINT A$ (X)
540  GOTO 90
550  END
```

Esta parte del programa compara las respuestas con el contenido de la memoria y en caso de error de la respuesta correcta.

El programa podría mejorarse borrando la pantalla **(CLS)** entre las instrucciones **490** y **495,** estableciendo un tiempo de espera (con **PAUSE, WAIT,** o instrucción similar según el ordenador) y lo mismo entre **530** y **540 (CLS** y **PAUSE, WAIT).**

15. Lengua e idiomas

1. Ejercicio de acentuación

Presentación del problema

Este programa da un texto literario de la obra *Platero y yo,* de Juan Ramón Jiménez, omitiendo los acentos. El alumno debe escribir las palabras que están mal escritas.

El texto es el siguiente:

"ACARICIE A PLATERO Y, COMO PUDE, LO ENGANCHE A LA CARRETILLA, DELANTE DEL BORRICO MISERABLE, LO OBLIGUE ENTONCES, CON UN CARIÑOSO IMPERIO, Y PLATERO, DE UN TIRON, SACO CARRETILLA Y RUCIO DEL ATOLLADERO, Y LES SUBIO LA CUESTA».

Las palabras solución son:

ACARICIÉ, ENGANCHÉ, OBLIGUÉ, TIRÓN, SACÓ, SUBIÓ

El programa

Lo primero que hay que hacer es almacenar el texto en memoria mediante una subrutina, para que aparezca en pantalla todas las veces que sea necesario. Las palabras solución se almacenan en una lista B$ de dimensión 6, pues hay seis palabras.

Se define una variable A$ para que el alumno dé sus respuestas.

Por último, se hace una comparación entre la respuesta A$ y la solución B$. Si es correcta, se pasa a la siguiente solución y, en caso contrario, se obliga a que el alumno responda otra vez. Este proceso se repite hasta que se conteste correctamente todas las palabras, con lo que el programa se termina.

```
 1 REM ACENTUACION
10 PRINT "DEBES DAR LAS PALABRAS QUE LLEVEN ACENTO, DE
   UNA EN UNA"
20 DIM B$(6)
30 B$(1)="ACARICIE"
40 B$(2)="ENGANCHE"
50 B$(3)="OBLIGUE"
60 B$(4)="TIRON"
70 B$(5)="SACO"
80 B$(6)="SUBIO"
```

En la instrucción **10** aparece lo que el alumno debe hacer durante la ejecución del programa. La instrucción **20** da la dimensión de 6 cadenas de caracteres. Por último, de la **30** a la **80** se introducen en la lista B$ las soluciones del problema.

```
 90  FOR I = 1 TO 6
 95  PRINT
100  GOSUB 500
110  INPUT A$
```

La instrucción **90** abre un bucle para las seis respuestas válidas. La **100** envía a la subrutina; ésta escribe el texto que se debe acentuar. La **110** sirve para que el alumno introduzca su respuesta.

```
160  IF A$=B$ (I) THEN GOTO 190
170  PRINT "ERROR; PRUEBA DE NUEVO"
180  GOTO 110
```

La instrucción **160** compara la respuesta del alumno A$ con la solución, B$ (I); si no es correcta, pasa el control a la instrucción siguiente (instrucción **170**), la cual le anuncia que es errónea la respuesta y que pruebe de nuevo; por ello, la instrucción **180** envía el control del programa a la **110**, para que el alumno introduzca otra respuesta (hasta que no sea correcta no se sale de este bucle). Si la respuesta es la correcta, la instrucción **160** envía el control a la **190.**

```
190  CLS
200  PRINT "CORRECTO, BUSCA LA PALABRA SIGUIENTE"
210  NEXT I
220  PRINT "NO BUSQUES MAS; HAS ENCONTRADO TODAS"
230  END
```

La **190** borra toda la pantalla y la **200** anuncia que la respuesta es correcta y que el alumno se prepare para la siguiente. En la **210** se termina el bucle comenzado en la instrucción **90.**

El fin del ejercicio se anuncia con la instrucción **220** y el programa termina con la instrucción **230 END** para que no se ejecuten las instrucciones siguientes, correspondientes a la subrutina.

```
500  PRINT "ACARICIE A PLATERO Y, COMO PUDE, LO ENGANCHE A
     LA CARRETILLA, DELANTE DEL BORRICO MISERABLE, LE OBLI-
     GUE ENTONCES, CON UN CARIÑOSO IMPERIO, Y PLATERO, DE
     UN TIRON, SACO CARRETILLA Y RUCIO DEL ATOLLADERO, Y
     LES SUBIO LA CUESTA".
510  RETURN
```

Estas dos últimas instrucciones componen la subrutina, llamada desde la instrucción **100.** La **500** escribe el texto de estudio, que se repetirá seis veces a lo largo de la ejecución del programa (una para cada respuesta), y la **510** envía el control a la instrucción siguiente: a la **110.**

2. Conjugación de verbos

Presentación del problema

Los verbos regulares se dividen en tres grupos, atendiendo a su terminación: en AR, en ER y en IR. La conjugación de todos los verbos de un mismo grupo es bastante sencilla pues sus terminaciones (desinencias) son iguales. Así, por ejemplo, para conjugar el presente de indicativo del verbo TARDAR basta con añadir a la parte radical TARD las terminaciones

O
AS
A
AMOS
AIS
AN

y anteponer los pronombres personales correspondientes:

YO	TARD	O
TU	TARD	AS
EL	TARD	A
NOSOTROS	TARD	AMOS
VOSOTROS	TARD	AIS
ELLOS	TARD	AN

El programa

El programa siguiente da el presente de indicativo, el pretérito imperfecto o copretérito, el pretérito indefinido o pretérito y el futuro imperfecto o futuro (todos estos tiempos forman el modo indicativo simple de un verbo de la primera conjugación, o sea de los terminados en AR).

Lo primero que hace el programa es pedir el verbo y almacenarlo en A$. A continuación calcula la longitud de A$ y extrae una subcadena B$, en la que falta la terminación AR.

Por último, se imprimen los distintos tiempos del modo indicativo simple.

```
  1 REM MODO INDICATIVO SIMPLE DE LA PRIMERA CONJUGACION
 10 INPUT "VERBO", A$
 20 L=LEN(A$)
 30 B$=LEFT$ (A$; L-2)
```

En la instrucción **10** se introduce el verbo, en la **20** se calcula su longitud y en la **30** se extrae una subcadena izquierda de longitud L-2. Luego B$ es la parte radical del verbo.

```
 40 PRINT "PRESENTE DE INDICATIVO"
 50 PRINT "YO", B$, "O"
 60 PRINT "TU", B$; "AS"
 70 PRINT "EL", B$; "A"
 80 PRINT "NOSOTROS", B$; "AMOS"
 90 PRINT "VOSOTROS", B$; "AIS"
100 PRINT "ELLOS", B$; "AN"
```

El bloque **40-100** imprime el presente de indicativo. Por ejemplo, la instrucción **50** imprime YO, salta a la siguiente zona (debido a la coma) y escribe B$ junto a O, pues al estar separados por punto y coma se imprimen seguidos.

```
110 PRINT "PRETERITO IMPERFECTO O CO-PRETERITO"
120 PRINT "YO", B$; "ABA"
130 PRINT "TU", B$; "ABAS"
140 PRINT "EL", B$, "ABA"
150 PRINT "NOSOTROS", B$; "ABAMOS"
160 PRINT "VOSOTROS", B$; "ABAIS"
170 PRINT "ELLOS", B$; "ABAN"
```

Este bloque imprime el pretérito imperfecto o co-pretérito.

```
180 PRINT "PRETERITO INDEFINIDO O PRETERITO"
190 PRINT "YO", B$; "E"
200 PRINT "TU", B$; "ASTE"
210 PRINT "EL", B$; "O"
220 PRINT "NOSOTROS", B$; "AMOS"
230 PRINT "VOSOTROS", B$; "ASTEIS"
240 PRINT "ELLOS", B$; "ARON"
```

Las instrucciones **180-240** imprimen el pretérito indefinido o pretérito.

```
250 PRINT "FUTURO IMPERFECTO O FUTURO"
260 PRINT "YO", A$; "E"
270 PRINT "TU", A$; "AS"
280 PRINT "EL", A$; "A"
290 PRINT "NOSOTROS", A$; "EMOS"
300 PRINT "VOSOTROS", A$; "EIS"
310 PRINT "ELLOS", A$; "AN"
```

En el bloque **250-310** se toma como parte radical el verbo completo A$, ya que la conjugación del futuro incluye en la parte radical la terminación AR.

3. Identificación de preposiciones inglesas

Planteamiento del problema

Se trata de que vayan apareciendo en pantalla una serie de frases en inglés, a las que le faltan las preposiciones. El que ejecute el programa deberá dar una preposición (respuesta) y la pantalla le comunicará si es correcta o no. Si lo es, pasará a la siguiente frase; en caso contrario, volverá a aparecer en la pantalla la misma frase.

Las frases son	Las preposiciones son
JOHN DREAMT ... MARY LAST NIGHT	OF
I'M LOOKING FORWARD ... MEETING YOU NEXT WEEKEND	TO
MARY IS THINKING ... JOHN AT THE MOMENT	OF
DON'T STAND THERE ... THE DOOR, COME ... AND SIT ...	AT-IN-DOWN
YOU CAN COUNT ... ME, I NEVER LET MY FRIENDS ...	ON-DOWN

El programa

El programa es abierto, es decir, sirve para cualquier número N de frases (en este caso N = 5) y no únicamente en inglés sino en cualquier otro idioma; basta con dar las frases y preposiciones en el idioma elegido.

La entrada de los datos, frases y preposiciones se realiza por medio de un bucle que tiene la instrucción READ A$, B$, almacenándose en A$ las frases y en B$ las preposiciones.

Previamente hay que dimensionar las listas A$ y B$.

El paso siguiente es definir dos variables Y$ y Z$ que, por medio de su impresión en pantalla, informen si la respuesta elegida es correcta o no.

Por último, el programa ha de contener un bucle en el que se dé una de las frases, se introduzca una respuesta y ésta sea comparada con la solución. No se pasará a la frase siguiente hasta que no se acierte la respuesta.

```
1  REM PREPOSICIONES EN INGLES
10 INPUT N
20 DIM A$(N) : DIM B$ (N)
30 FOR I = 1 TO N
40 READ A$(I), B$(I)
50 NEXT I
60 Y$ = "NO ES CORRECTA, REPITE"
70 Z$ = "CORRECTA, AQUI TIENES OTRA FRASE"
```

En la instrucción **10** se introduce el número de frases que hay. La **20** dimensiona las N frases y las N preposiciones. De la **30** a la **50,** el bucle lee las frases y las preposiciones. Por último, en la **60** y **70** se definen dos variables, Y$ y X$, que servirán para decir al alumno si ha acertado o no.

```
80  FOR I = 1 TO N
90  PRINT "PON LA PREPOSICION QUE FALTA Y SI HAY VARIAS
    DEJA UN ESPACIO ENTRE ELLAS"
100 PRINT A$(I)
110 INPUT X$
```

En la instrucción **80** comienza la serie de instrucciones que llevarán al alumno a contestar las distintas cuestiones. En la **90** se le dice qué es lo que debe responder y cómo, en el caso de varias preposiciones. La instrucción **100** le da la frase que tiene que completar con la preposición o preposiciones correctas. Por último, en la **110** se le pide la respuesta X$.

```
160 IF X$ = B$(I) THEN GOTO 190
170 PRINT Y$
180 GOTO 100
```

La instrucción **160** compara la solución B$ con la respuesta X$ dada por el alumno; si no es correcta, la **170** le dice que repita, por medio de la variable Y$ definida en **60;** y la **180** devuelve el control a la instrucción **100,** con lo que se da la oportunidad al alumno de repetir su respuesta X$. Este proceso no termina mientras no dé la respuesta correcta.

```
190  CLS
200  PRINT Z$
210  NEXT I
220  PRINT "¡ENHORABUENA! LLEGASTE A LA META"
```

A la instrucción **190** se llega cuando el alumno ha acertado la preposición correspondiente; esta instrucción borra toda la pantalla. Posteriormente, se da al alumno, por medio de la variable Z$ (en la instrucción **200**), la información de haber acertado con su respuesta. Mediante la instrucción **210** se repite el ciclo con otra frase. Con la instrucción **220** se le comunica al alumno el fin de las frases y, por tanto, la terminación del programa.

Este programa sirve para cualquier número de frases, pero si se quiere utilizar para el ejemplo presentado, hay que introducir los siguientes datos:

```
230  DATA 5
240  DATA "JOHN DREAMT ... MARY LAST NIGHT", "OF"
250  DATA "I'M LOOKING FORWARD ... MEETING YOU NEXT WEE-
     KEND", "TO"
260  DATA "MARY IS THINKING ... JOHN AT THE MOMENT", "OF"
270  DATA "DON'T STAND THERE ... THE DOOR, COME ... AND SIT ...",
     "AT IN DOWN"
280  DATA "YOU CAN COUNT ... ME, I NEVER LET MY FRIENDS ...",
     "ON DOWN"
290  END
```

4. Repasando la morfología y sintaxis latina

Representación del problema

En el texto siguiente:

"ACEPT ORATIONE EORUM, CAESAR OBSIDES TRADI IMPERAT EOSQUE ADUCCI IUBET. HIS ADDUCTIS, UT IMPERAVERAT, ARBITROS INTER CIVITATES DAT, QUI LITEM AESTIMENT"

el alumno debe buscar los sustantivos (excluidos los nombres propios) y, una vez encontrados todos, ponerlos en el mismo caso, pero en número distinto al que tienen en el texto.

Las soluciones son:

ORATIONE	ORATIONIBUS
OBSIDES	OBSIDEM
ARBITROS	ARBITRUM
CIVITATES	CIVITATEM
LITEM	LITES
	(CAESAR ES NOMBRE PROPIO)

El programa

Se utilizan dos variables B$(I) y C$(I) para introducir en la memoria del microordenador las palabras solución.

El texto, incluido en dos variables T$ y U$, se almacena en una subrutina para no sobrepasar el límite de 255 caracteres.

En un bucle se comprueba si cada respuesta es correcta. Cuando se han encontrado todos los sustantivos, se pide que se pongan en distinto número. De nuevo, las respuestas se comparan con las palabras solución y en caso de equivocación se da el resultado correcto.

```
 1 REM LATIN
10 READ N
20 DIM B$ (N)
25 DIM C$ (N)
30 FOR I = 1 TO N
40 READ B$ (I)
50 READ C$ (I)
60 NEXT I
```

En estas instrucciones (en combinación con las instrucciones **DATA** del final del programa) se introducen los sustantivos en distintos números C$ (I).

```
70  GOSUB 500
80  PRINT "BUSCA TODOS LOS SUSTANTIVOS EN EL TEXTO SI-
    GUIENTE"
90  INPUT A$
100 FOR I = 1 TO N
110 IF A$ = B$ (I) THEN GOTO 130
120 GOTO 160
130 PRINT "ACERTASTE"
140 C = C + 1
150 GOTO 190
160 NEXT I
170 PRINT "TE EQUIVOCASTE, BUSCA DE NUEVO"
180 GOTO 90
190 IF C < N THEN GOTO 90
200 PRINT "YA HAS ENCONTRADO TODOS"
```

Una vez puesto el texto en pantalla (instrucción **70 GOSUB 500)** se piden los sustantivos (instrucciones **80** y **90)** y en el bucle siguiente (instrucciones **100** a **160**) se comprueba si la palabra dada es alguna de las almacenadas en la tabla B$ (I); si no es, se pide una nueva respuesta, y si es, se suma uno al contador C. Cuando C coincide con el número de sustantivos existentes (instrucciones **160** a **170**) se imprime un mensaje.

```
210 FOR K = 1 TO 2000 : NEXT K
220 CLS
230 PRINT "PON LOS SUSTANTIVOS ANTERIORES EN EL MISMO
    CASO PERO EN DISTINTO NUMERO"
240 FOR I = 1 TO N
250 PRINT B$ (I)
260 INPUT D$
270 IF D$ = C$ (I) THEN GOTO 300
280 PRINT "TE EQUIVOCASTE, LA RESPUESTA ES:"; C$ (I)
290 GOTO 310
300 PRINT "DE ACUERDO"
310 NEXT I
320 STOP
```

Después de un tiempo de espera (instrucción **210**), se borra la pantalla (instrucción **220**) y se hace una nueva pregunta **(230).** La respuesta se compara con el contenido de la memoria C$ (I).

```
500 T$ = "ACCEPTA ORATIONE EORUM, CAESAR OBSIDES TRADI
    IMPERAT EOSQUE ADUCCI IUBET"
510 U$ = "HISADDUCTIS, UT IMPERAVERAT, ARBITROS INTER
    CIVITATES DAT, QUI LITEM AESTIMENT"
520 PRINT T$
530 PRINT U$
540 RETURN
```

Con la subrutina **(500-540)** se almacena el texto en dos variables, T$ y U$, y a continuación se imprime

```
590 DATA 5
600 DATA ORATIONE, ORATIONIBUS, OBSIDES, OBSIDEM, ARBI-
    TROS, ARBITRUM
610 DATA CIVITATES, CIVITATEM, LITEM, LITES
620 END
```

En las instrucciones **DATA** se incluyen las distintas soluciones.

16. Control de una pequeña biblioteca

Presentación del problema

El almacenamiento y el tratamiento de información se hace habitualmente con dispositivos que utilizan discos o disquetes. Sin embargo, si la cantidad de información no ocupa excesiva memoria, se puede utilizar un simple grabador, reproductor de casetes.

El programa

El programa siguiente sirve para controlar una pequeña biblioteca, para lo cual permite llevar a cabo las siguientes acciones:

— Crear el archivo de libros.

— Cargar el archivo desde casete.

— Añadir libros al archivo ya creado.

— Buscar libros de una materia.

— Buscar libros de un autor.

— Buscar el autor de un título.

Para construir el programa se define la variable L$ (I) mediante la concatenación de la variable A$ (I) que incluye un número clave con dos cifras (correspondiente a la materia), la variable B$ (I) con un número clave (correspondiente a la submateria), la variable C$ (I) que incluye el autor de la obra (para el que se reservan 30 caracteres) y la variable D$ (I) que incluye el título del libro (para el que también se reservan 30 caracteres).

Una vez ordenados los libros introducidos, tomando diversos fragmentos de la variable L$ (I), con la ayuda de las instrucciones **LEFT$ ( ), MID$ ( )** y **RIGHT$ ( )** se puede localizar materia, autor o título según se desee.

```
1 REM BIBLIOTECA
10 CLEAR 3000
20 DIM A$ (50), B$ (50), C$ (50), D$ (50), E$ (50), L$ (50)
30 DIM C(50), D(50)
40 CLS
50 PRINT "BIBLIOTECA"
60 PRINT : PRINT : PRINT
70 PRINT "ESCRIBE:"
80 PRINT
90 PRINT "1 - PARA CREAR ARCHIVO"
100 PRINT "2 - PARA MODIFICAR ARCHIVO"
110 PRINT "3 - PARA CARGAR ARCHIVO"
120 PRINT "4 - PARA BUSCAR LIBROS DE UNA MATERIA"
130 PRINT "5 - PARA BUSCAR LIBROS DE UN AUTOR"
140 PRINT "6 - PARA BUSCAR EL AUTOR DE UN TITULO"
150 PRINT "7 - PARA ACABAR"
160 INPUT A
170 IF A < 1 OR A > 7 THEN GOTO 40
180 ON A GOSUB 1000, 2000, 3000, 4000, 5000, 6000, 7000
190 GOTO 40
```

La instrucción **10** limpia la parte de memoria correspondiente a las cadenas, lo que permite introducir el número de caracteres necesarios.

El resto de las instrucciones sirve para reservar posiciones de memoria, y presenta las opciones disponibles. Cada una de estas opciones corresponde a una subrutina diferente.

```
1000 REM ARCHIVO
1010 CLS
1020 I = 1
1030 LOCATE 1,5 : INPUT "CLAVE"; A$ (I)
1040 LOCATE 1,7 : INPUT "SUBCLAVE": B$ (I)
1050 LOCATE 1,9 : INPUT "AUTOR"; C$ (I)
1060 LOCATE 1,11 : INPUT "TITULO"; D$ (I)
1070 LOCATE 1,20 : INPUT "DE ACUERDO (S/N)"; F$
1080 IF F$ = "N" OR F$ = "n" THEN GOTO 1030
```

Con estas instrucciones se introducen los datos correspondientes a cada libro.

```
1100 C (I) = LEN (C$(I))
1110 D(I) = LEN(D$(I))
1120 FOR J = C(I) + 1 TO 30
1130 C$ (I) = C$ (I) + ""
```

```
1140  NEXT J
1150  FOR K = D (I) + TO 30
1160  D$ (I) = D$ (I) + ""
1170  NEXT K
```

Aquí añaden espacios en blanco a C$ (I) y D$ (I) hasta que alcancen una longitud de 30.

```
1180  L$ (I) = A$ (I) + B$ (I) + C$ (I) + D$ (I)
1190  INPUT "MAS LIBROS"; F$
1200  IF F$ = "N" OR F$ = "n" THEN GOTO 1300
1210  I = I + 1
1220  GOTO 1030
```

Este bloque de instrucciones concatena las cadenas A$ (I), B$ (I), C$ (I) y D$ (I), formando la cadena L$ (I). Además permite introducir un nuevo libro, en cuyo caso se incrementa la variable I en una unidad y a continuación se reinicia el proceso.

```
1300  FOR J = 1 TO I - 1
1310  FOR K = J + 1 TO I
1320  IF L$ (J) < L$ (K) THEN GOTO 1370
1330  X$ = L$ (J)
1340  L$ (J) = L$ (K)
1350  L$ (K) = X$
1360  NEXT K
1370  NEXT J
```

En este grupo de instrucciones se ordenan las variables L$ (I).

```
1400  CLS
1410  PRINT "PREPARAR EL CASETE PARA GRABAR"
1415  PRINT : PRINT : INPUT "PREPARADO S/N"; F$
1420  IF F$ = "N" OR F$ = "n" THEN GOTO 1400
1430  OPEN "CAS : LIBR" FOR OUTPUT AS # 1
1440  FOR J = 1 TO I
1450  PRINT # 1, L$ (J)
1460  NEXT J
1470  CLOSE # 1
1480  PRINT "PULSA UNA TECLA"
1490  A$ = INKEY$
1500  IF A$ = "" THEN GOTO 1490
1510  RETURN
```

Mediante estas instrucciones, los datos almacenados en L$ (I) se graban en un casete. Las instrucciones **1480** y **1510** permiten volver al punto de partida.

```
2000  REM MODIFICAR
2010  CLS
2020  GOTO 1030
```

Volviendo a la instrucción **1030** pueden añadirse libros a un archivo ya creado.

```
3000  REM CARGAR
3005  CLS
3010  PRINT "PREPARAR EL CASETE PARA CARGAR"
3020  PRINT : PRINT : PRINT : INPUT "PREPARADO S/N"; F$
3030  IF F$ = "N" OR F$ = "n" THEN GOTO 3010
3040  OPEN "CAS : LIBR" FOR INPUT AS # 1
3050  I = 1
3060  IF EOF(1) = −1 THEN CLOSE #1 : RETURN
3070  INPUT # 1, L$ (I)
3080  I = I + 1
3090  GOTO 3060
```

Se van leyendo libros desde el casete, hasta que en la instrucción **3060** se detecta el final del archivo, pasándose entonces al menú principal.

```
4000  REM MATERIA
4005  CLS
4010  INPUT "INTRODUZCA LA CLAVE DE UNA MATERIA"; H$
4020  FOR J = 1 TO I
4030  IF H$ = LEFT$ (L$(J), 2) THEN PRINT LEFT$(L$(J), 3) : PRINT
      MID$ (L$(J), 4,30) : PRINT RIGHT$(L$(J), 30)
4040  NEXT J
4050  PRINT "PULSA UNA TECLA PARA VOLVER"
4060  A$ = INKEY$
4070  IF A$ = "" THEN 4060
4080  RETURN
```

Se localizan los libros correspondientes a una clave determinada.

```
5000 REM AUTOR
5005 CLS
5010 INPUT "INTRODUCE EL NOMBRE DEL AUTOR"; N$
5020 U = LEN (N$)
5030 FOR X = U + 1 TO 30
5040 N$ = N$ + ""
5050 NEXT X
5060 FOR J = 1 TO I
5070 IF MID$ (L$(J), 4,30) = N$ THEN PRINT LEFT$ (L$(J), 3) : PRINT
     MID$ (L$ (J), 4,30) : PRINT RIGHT$ (L$ (J), 30)
5080 NEXT J
5090 PRINT "PULSA UNA TECLA PARA VOLVER"
5100 A$ = INKEY$
5110 IF A$ = "" THEN GOTO 5100
5120 RETURN
```

Se introduce el nombre de un autor al que una vez determinada su longitud se le añaden espacios en blanco hasta completar 30 carácteres. El bucle comprendido entre **5060** y **5080** localiza el autor en la cadena L$(I).

```
6000 REM TITULO
6010 CLS
6020 INPUT "INTRODUZCA TITULO"; T$
6030 V = LEN(T$)
6040 FOR R = V + TO 30
6050 T$ = T$ + ""
6060 NEXT R
6070 FOR J = 1 TO I
6080 IF T$ = RIGHT$(L$(J), 30) THEN PRINT LEFT$(L$(J), 3) : PRINT
     MID$(L$(J), 4,30) : PRINT RIGHT$(L$(J), 30)
6090 NEXT J
6100 PRINT "PULSA UNA TECLA"
6110 A$ = INKEY$
6120 IF A$ = "" THEN GOTO 6110
6130 RETURN 40
```

Con este último bloque de instrucciones se introduce un título, se completa hasta 30 caracteres y, en caso de localizarse entre los existentes, se imprimen la clave, subclave, el autor y el propio título.

```
7000 REM FIN
7010 END
```

17. Un programa para escribir y otro para dibujar

1. Programa para escribir

El programa siguiente permite llevar a cabo estas acciones:

— Escribir una página de 20 líneas con 38 caracteres cada una.
— Ver una página completa en la pantalla.
— Guardar una página en un casete.
— Reproducir una página que se ha grabado previamente en un casete.

```
  1 REM ESCRIBIR
 10 CLEAR 3000
 20 DIM D$(22, 40)
 30 CLS
 40 PRINT "ESCRIBE :"
 50 PRINT "1-PARA ESCRIBIR PAGINA"
 60 PRINT "2-PARA VER PAGINA"
 70 PRINT "3-PARA GUARDAR PAGINA EN CASETE"
 80 PRINT "4-PARA REPRODUCIR PAGINA DE CASETE"
 90 PRINT "5-PARA ACABAR"
100 INPUT A
110 IF A < 1 OR A > 5 THEN GOTO 30
120 ON A GOSUB 1000, 2000, 3000, 4000, 5000
130 GOTO 30
```

Con estas instrucciones se reserva espacio en la memoria y se presentan las opciones de que dispone el programa.

```
1000 REM ESCRIBIR
1005 PRINT "ESCRIBA/PARA ACABAR" : FOR K = 1 TO 2000 : NEXT K
1010 CLS
1020 I = 1 : J = 1
1030 IF J < 1 OR J = 38 THEN J = 1
1040 A$ = INKEY$
1050 IF A$ = ""THEN GOTO 1040
1055 IF A$ = "" THEN CLS : RETURN
1060 IF ASC(A$) = 13 THEN J = 38; GOTO 1110
1070 IF ASC(A$) = 8 THEN PRINT CHR$(29); ""; CHR$(29); : J = J -
     1 : D$ (I, J) = "" : GOTO 1030
1080 D$(I, J) = A$ : PRINT D$(I, J);
1085 J = J + 1
1090 IF J = 38 THEN GOTO 1110
1100 GOTO 1040
1110 PRINT
1115 IF I = 20 THEN CLS : RETURN
1120 I = I + 1
1130 GOTO 1030
```

En la instrucción **1040** se almacenan en la cadena A$ los caracteres que se van pulsando.

Cuando se pulsa la tecla RETURN (código 13) se produce un salto de línea. Cuando se pulsa la tecla BS (**DELETE** o similar), con código 8, se borra el carácter que se acaba de escribir, ya que **PRINT CHR$(29)** hace que el cursor retroceda.

Por otra parte, en la cadena D$(I, J) se van almacenando los distintos caracteres pulsados.

```
2000 REM VER PAGINA
2005 PRINT "CUANDO ACABES PULSA UNA TECLA PARA SEGUIR" :
     FOR K = 1 TO 1000 : NEXT K
2010 CLS
2015 FOR I = 1 TO 20
2020 FOR J = 1 TO 38
2030 PRINT D$(I, J);
2040 NEXT J
2045 PRINT
2050 NEXT I
2070 A$ = INKEY$
2080 IF A$ = "" THEN GOTO 2070
2090 CLS
2100 RETURN
```

Los caracteres almacenados en D$(I, J) se imprimen en la pantalla. Al pulsar una tecla se vuelve al menú inicial.

```
3000  REM GUARDAR PAGINA
3010  CLS
3020  PRINT "PREPARAR EL CASETE PARA GRABAR"
3030  PRINT : PRINT : INPUT "PREPARADO(S/N)"; F$
3040  IF F$ = "N" OR F$ = "n" THEN GOTO 3020
3050  OPEN "CAS : PAG" FOR OUTPUT AS #1
3060  FOR I = 1 TO 20
3070  FOR J = 1 TO 38
3080  PRINT # 1, D$(I, J)
3090  NEXT J
3100  NEXT I
3110  CLOSE # 1
3120  RETURN
```

Los caracteres almacenados en D$(I, J) se almacenan en un casete.

```
4000  REM CARGAR
4010  CLS
4020  PRINT "PREPARAR EL CASETE PARA PRODUCIR"
4030  INPUT "PREPARADO(S/N)"; F$
4040  IF F$ = "N" OR F$ = "n" THEN GOTO 4020
4050  OPEN "CAS : PAG" FOR INPUT AS # 1
4060  FOR I = 1 TO 20
4070  FOR J = 1 TO 38
4080  INPUT # 1, D$(I, J)
4090  NEXT J
4100  NEXT I
4110  CLOSE # 1
4120  RETURN
```

Los caracteres almacenados en un casete se introducen en la memoria del microordenador en las variables D$(I, J). Para ver la página puede utilizarse a continuación la opción 2.

```
5000  REM ACABAR
5010  END
```

2. Programa para dibujar

El programa siguiente permite realizar estas acciones:

— Dibujar en la pantalla con la ayuda de un joystick (moviéndolo en ocho direcciones) o, inicialmente, usando las teclas de movimiento del cursor. Borrar cuando sea necesario y almacenar en la memoria del microordenador los puntos de un dibujo que hayan resultado satisfactorios.

— Ver un dibujo en la pantalla cuando los puntos que van a ayudar a trazarlo están almacenados en la memoria.

— Guardar en un casete un dibujo que apareció en la pantalla.

— Cargar de un casete los puntos necesarios para reproducir un dibujo en la pantalla de un microordenador.

```
 10 DIM X(100), Y(100)
 20 CLS
 30 PRINT "DIBUJAR"
 40 PRINT "1-PARA DIBUJAR"
 50 PRINT "2-PARA VER DIBUJO"
 60 PRINT "3-PARA GUARDAR DIBUJO EN CASETE"
 70 PRINT "4-PARA CARGAR DIBUJO EN CASETE"
 80 PRINT "5-PARA ACABAR"
 90 INPUT A
100 IF A < 1 OR A > 5 THEN GOTO 20
110 ON A GOSUB 1000, 2000, 3000, 4000, 5000
120 GOTO 20
```

Con este bloque de instrucciones se reserva espacio en la memoria para almacenar las coordenadas de los puntos que formarán el dibujo. Se presentan después las opciones de que dispone el programa.

```
1000 REM DIBUJAR
1010 CLS
1020 PRINT "ESCRIBE D : DIBUJAR"
1030 PRINT "ESCRIBE B : BORRAR"
1040 PRINT "ESCRIBE V : GUARDAR PUNTO"
1050 PRINT "ESCRIBE S PARA SALIR"
1060 PRINT "PULSA UNA TECLA"
1070 C$ = INKEY$
1080 IF C$ = "" THEN GOTO 1070
1090 SCREEN 2
1100 X = 128 : Y = 87 : L = 0 : I = 1
1110 A$ = INKEY$ : A = STICK (0)
```

```
1120  IF A$ = "" AND A = 0 THEN GOTO 1110
1130  IF L = 0 THEN PSET(X, Y)
1140  IF L = 1 THEN PRESET(X, Y)
1150  IF L = 2 THEN X(J) = X : Y(J) = Y : J = J + 1 : PSET (X, Y) : L = 0
1160  IF A = 1 THEN Y = Y - 1
1170  IF A = 2 THEN Y = Y - 1 : X = X + 1
1180  IF A = 3 THEN X = X + 1
1190  IF A = 4 THEN X = X + 1 : Y = Y + 1
1200  IF A = 5 THEN Y = Y + 1
1210  IF A = 6 THEN Y = Y + 1 : X = X - 1
1220  IF A = 7 THEN X = X - 1
1230  IF A = 8 THEN X = X + 1 : Y = Y - 1
1240  IF X > 255 THEN X = 255
1250  IF X < 1 THEN X = 1
1260  IF Y < 1 THEN Y = 1
1270  IF Y < 190 THEN Y = 190
1280  IF A$ = "B" THEN L = 1
1290  IF A$ = "D" THEN L = 0
1300  IF A$ = "V" THEN L = 2
1310  IF A$ = "S" THEN SCREEN 0 : RETURN
1340  GOTO 1110
```

Con la ayuda de las teclas del cursor (STICK(0)) o con el joystick (STICK(1)) o (STICK(2)) se desplaza el cursor en una pantalla de alta resolución.

Si previamente se ha pulsado D, el cursor deja huella de su paso, y si se ha pulsado B, dicha huella queda borrada.

Cuando se pulsa V, se almacena en las variables X(I) e Y(I) el punto donde se encuentra el cursor. Pulsando S el proceso finaliza.

```
2000  REM VER DIBUJO
2010  CLS
2020  PRINT "PULSA S PARA SALIR"
2030  FOR K = 1 TO 1000 : NEXT K
2040  SCREEN 2
2050  CLS
2060  FOR I = 0 TO J
2070  IF X(I + 1) = 0 AND Y(I + 1) = 0 THEN X(Y + 1) = X(I) : Y(Y
      + 1) = Y(I)
2080  LINE(X(I), Y(I)) - (X(I + 1), Y(I + 1))
2090  NEXT I
2100  A$ = INKEY$
2110  IF A$ = "S" THEN CLS : CLOSE # 1 : SCREEN 0 : RETURN 20
2120  GOTO 2100
```

Los puntos almacenados (después de haber pulsado V) se unen entre sí en la pantalla del microordenador mediante tramos rectos. Este proceso se termina cuando se llega a una coordenada nula para X e Y.

```
3000  REM GUARDAR DIBUJO
3010  CLS
3020  PRINT "PREPARAR EL CASETE PARA GRABAR"
3030  PRINT : PRINT : INPUT "PREPARADO (S/N)"; F$
3040  IF F$ = "N" OR F$ = "n" THEN GOTO 3020
3050  OPEN "CAS : DIB" FOR OUTPUT AS # I
3060  FOR I = 0 TO J
3070  PRINT # 1, X(I)
3080  PRINT # 1, Y(I)
3090  NEXT I
3100  CLOSE 1
3120  PRINT "PULSA UNA TECLA"
3130  A$ = INKEY$
3140  IF A$ = "" THEN GOTO 3130
3150  CLS
3160  RETURN
```

Las coordenadas X(I), Y(I) almacenadas en la memoria del microordenador se graban en un casete.

```
4000  REM CARGAR DIBUJO
4010  CLS
4020  PRINT "PREPARA EL CASETE PARA REPRODUCIR"
4030  PRINT : PRINT : INPUT "PREPARADO(S/N)"; F$
4040  IF F$ = "N" OR F$ = "n" THEN GOTO 4020
4050  OPEN "CAS : DIB" FOR INPUT AS # 1
4060  I = 1
4070  IF EOF(1) = –1 THEN CLOSE # 1 : RETURN
4080  INPUT # 1, X(I)
4090  INPUT # 1, Y(I)
4100  I = I + 1
4110  GOTO 4070
```

Desde un casete se llevan a la memoria del microordenador las coordenadas de los puntos de un dibujo, y dichas coordenadas se almacenan en las variables X(I) e Y(I).

Cuando en la instrucción **4070** se detecta el final del archivo, acaba todo el proceso.

```
5000  REM ACABAR
5010  END
```

COLECCIÓN BASIC

Basic Programación

Gráficos, Colores y Música en el ZX Spectrum

MSX. Programación con Gráficos, Colores y Música

Programas resueltos en BASIC

Programas de aplicaciones en BASIC

distribuidor exclusivo
cesma sa
C/ Aguacate, 25
28044 MADRID